Department of AERONAUTICS and ASTRONAUTICS
STANFORD UNIVERSITY

SUDAAR 595

EXPERIMENTS IN THRUSTERLESS ROBOT LOCOMOTION CONTROL FOR SPACE APPLICATIONS

Warren J. Jasper

Department of Aeronautics and Astronautics
Stanford University
Stanford, California 94305

Research supported by NASA contract NCC 2-333

September 1990

EXPERIMENTS IN THRUSTERLESS ROBOT LOCOMOTION CONTROL FOR SPACE APPLICATIONS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Warren Joseph Jasper

September 1990

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Robert H. Cannon, Jr.
Department of Aeronautics and Astronautics
(Principal Adviser)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Gene F. Franklin
Department of Electrical Engineering

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Daniel B. DeBra
Department of Aeronautics and Astronautics

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Abstract

While performing complex assembly tasks or moving about in space, a space robot should minimize the amount of propellant consumed. This thesis comprises an analytical and experimental study of space robot locomotion and orientation without the use of thrusters. The goal of this research is to design a robust control paradigm that will perform thrusterless locomotion between two points on a structure, and to implement this paradigm on an experimental robot.

A two-arm free-flying robot has been constructed which floats on a cushion of air to simulate in two dimensions the drag-free, zero-g environment of space. The robot can impart momentum to itself by pushing off from an external structure in a coordinated two-arm maneuver, and can then reorient itself by activating a momentum wheel.

The controller design consists of two parts: a high-level strategic controller and a low-level dynamic controller. The strategic controller, implemented as a finite-state machine, monitors the state of the system and switches control laws asynchronously, based on discrete events. Different specific control laws are implemented depending upon the configuration of the system, the number of degrees of freedom, and the desired task. The dynamic controller consists of a system of estimators, control laws, trajectory generators, and filters. For example, whenever both arms are grasping an external structure, the strategic controller installs a momentum controller which causes the linear and angular momentum of the *system* to follow desired trajectories.

The control paradigm has been verified experimentally by commanding the robot to push off from a structure with both arms, rotate 180 degrees while translating freely, and then catch itself on another structure. This method, based on the idea of *computed torque*, provides a linear feedback law in momentum and its derivatives for a system of rigid bodies. By controlling momentum, a configuration-independent quantity, the robot can leap precisely from one place to another, while accounting for nonlinear forces and changing kinematic constraints. It is believed that this design approach can be easily extended to three dimensions and to more complex robot configurations.

To Noreen and Steve

Acknowledgements

I wish to thank my principal advisor, Professor Robert H. Cannon Jr., for his enthusiastic support and guidance in this research, and for creating an environment at the Stanford Aerospace Robotics Laboratory that fosters creativity and promotes fundamental research in robotics. The atmosphere in the lab has enriched my stay at Stanford, and provides a high standard on how research laboratories should be run. I would also like to thank Professors Gene Franklin, Daniel DeBra, and Stephen Rock for their thorough review and constructive criticisms of my work.

The successful outcome of this experimental thesis resulted from the hard work and dedication of many gifted people. I am grateful to Gad Shelef for the design and fabrication of the mechanical hardware; and to Yosi Drucker, Joseph Schlesinger, and Godwin Zhang for their efforts in the design and fabrication of the electronics.

My thanks go to my fellow students Marc Ullman and Ross Koningstein for the initial design of the robot and many thoughtful discussions; Stan Schneider for implementing a useful software environment for real-time control; Vince Chen and Edward Wilson for their help and comments for sensor calibration; Robert Zanutta for his invaluable help in fabrication and modelling; and Dennis Morse for teaching me the finer points of Unix system administration. I also want to thank all the other graduate students in the Aerospace Robotics Laboratory for their help and suggestions during my five years in the lab.

This research was funded by the National Aeronautics and Space Administration under contract NCC 2-3333. Fellowship support was also given by the Hughes Aircraft Company.

Finally, I would like to thank my parents, William and Retha Jasper, for their loving support and commitment to higher education. I am also grateful to my sister and brother, Noreen and Steve, for their understanding and encouragement. It is to them that I dedicate this dissertation.

Contents

List of Tables

List of Figures

List of Symbols

The air bearing

μ	dynamic viscosity (18.0 μN/m^2)
g	acceleration due to gravity (9.8 m/s^2)
h	thickness of air gap (50.8 μm nominal)
m	total mass of vehicle (75.78 kg)
r_i	plenum radius (50.8 mm)
r_o	base radius (241.3 mm)
P	pressure on the plate
Q	steady state flow rate through air bearing
D	viscous drag on the air bearing

State variables and coordinates

q_i	generalized coordinate
u_i	generalized speed
s_i	abbreviation for $\sin(q_i)$
s_{ij}	abbreviation for $\sin(q_i + q_j)$

c_i	abbreviation for $\cos(q_i)$
c_{ij}	abbreviation for $\cos(q_i + q_j)$
m_i	mass of body i
L_{ij}	length of body j of link i
${}^A\boldsymbol{a}^{P_i}$	acceleration vector of point P_i in reference frame A
${}^A\boldsymbol{v}^{P_i}$	velocity vector of point P_i in reference frame A
${}^A\boldsymbol{\omega}^B$	angular velocity vector of body B in reference frame A
${}^A\boldsymbol{v}_r^{P_i}$	the r$^{\text{th}}$ partial velocity vector of point P_i in reference frame A
${}^A\boldsymbol{\omega}_r^B$	the r$^{\text{th}}$ partial angular velocity vector of body B in reference frame A
$\boldsymbol{f}_{tip}$	$4 \times n$ vector of forces at the tips of the arm
$\boldsymbol{A}_{rs}$	nonholonimic constraint matrix
$\boldsymbol{M}$	$n \times n$ Mass Matrix
$\tilde{\boldsymbol{M}}$	constrained mass matrix
$\boldsymbol{T}$	tranformation matrix from torques to generalized forces
$\tilde{\boldsymbol{T}}$	constrained tranformation matrix from torques to generalized forces
$\boldsymbol{Y}$	$n \times n$ transformation matrix from $\dot{\boldsymbol{q}}$ to $\boldsymbol{u}$
$\boldsymbol{c}$	vector of Coriolis and centrifugal forces
$\tilde{\boldsymbol{c}}$	constrained vector of Coriolis and centrifugal forces
$\boldsymbol{L}$	linear momentum vector
$\boldsymbol{H}$	angular momentum vector

Acronyms

ACV	Air Cushion Vehicle
ARL	Stanford University Aerospace Robotics Laboratory
CMG	Control Moment Gyros
DAC	Digital to Analog Converter
DOF	Degrees of Freedom
DSP	Digital Signal Processor
EKF	Extended Kalman Filter
EOM	Equations of Motion
FLOPS	Floating-Point Operations Per Second (a measure of computer speed)
GNU	Gnu Not Unix, Acronym of the Free Software Foundation
GPS	Global Positioning System
LED	Light Emitting Diode
MTBF	Mean Time Between Failure
PID	Proportional Integral Derivative error feedback control
RVDT	Rotary Variable Differential Transformer
SCARA	Selective Compliance Assembly Robot Arm
SCFH	Standard Cubic foot per Hour
SMD	Surface Mounted Device
SRMS	Space Shuttle Remote Manipulator System
ZOH	Zero Order Hold

Chapter 1

Introduction

This dissertation describes the LEAP (Locomotion Enhancement via Arm Push-off) project, conducted at The Stanford University Aerospace Robotics Laboratory (ARL) during the period 1986-1990.

1.1 Motivation

Much of the recent work in robotics has assumed *a priori* that the desired task is within the workspace of the robot. This assumption is clearly evident for all fixed-base robots, whether they be bolted to the factory floor or to the cargo bay of the space shuttle. A fixed-base robot is ideally suited to perform repetitive tasks in a highly structured environment.

Unfortunately, this is not the situation in space, where the environment is unstructured and the tasks are varied. For example, building or repairing a space-station requires the ability to perform a myriad of different tasks at different locations. Many missions that the astronauts are called upon to perform require mobility, such as repairing a damaged satellite. Therefore, a space robot must be mobile to perform different tasks at various locations. If space robots are to be mobile, then one must ask the question: "What are good ways for a robot to move from one place to another in space".

The ways a robot can move through space can be divided into two types: Those that use propulsion as the primary means for imparting momentum to the robot, and those that do not. For these two types, this section will briefly examine the issues involved in imparting momentum to a space vehicle and reorienting it.

Thrusters as a primary means of locomotion Historically, thrusters have been the only method employed to relocate space vehicles. The reason for their exclusive use is because expelling propellant is the only way for a totally isolated vehicle to impart momentum to itself. This capability comes with three disadvantages, namely: Propellant is a finite resource that must be transported into space; the propellant may impact or contaminate the surrounding environment, damaging sensitive equipment; and use of nonlinear on-off thrusters limits the degree of achievable precision in pointing and positioning.

On the other hand, in a multi-body environment, one can reduce or eliminate thrusters by pushing off from and landing on different structures. Thrusters then need only be employed for mid-course correction or during emergency situations, thereby reducing the total amount of propellant consumed. This is a very natural and intuitive approach to space locomotion, and is the way humans move around in space.

Thrusterless Attitude Control Thrusterless attitude control schemes use conservation of angular momentum to change orientation without imparting angular momentum to the system. This principle was first discovered by Newton [23] around 1666. One of the earlist references for using this principle in spacecraft orientation is due to Hohmann [12] in 1925. Hohmann advocates using only one thruster on a spacecraft to save weight. To demonstrate that a single thruster can be oriented in any direction, there is a picture of two astronauts crawling along a circular ladder to change the spacecraft's orientation.

Since the early 70's, satellites have used momentum wheels and magnetic coils to control spacecraft attitude [38]. While these techniques can control only the attitude (and not the location) of the vehicle, they do so with the advantage of converting electrical energy [1] into torque without using gas propellants. Longman [20] explores the possibility of using reaction wheels on a space vehicle with a robotic arm to stabilize vehicle attitude only. In this way, the base of the robot arm can be treated as inertially fixed, although Longman does not compensate for the linear motion of the base due to the motion of the arm.

A unified approach to space robotic locomotion needs to address two issues: controlling the *location of the mass center* of the robot, and controlling the *orientation about the mass center* of the robot. The first issue involves controlling linear momentum, whereas the

[1] Electrical energy is obtainable from the sun by photo voltaic cells, and is therefore a replenishable resource

second issue involves controlling the geometric configuration of the robot. These issues will be dealt with in detail throughout this thesis.

1.2 Research Goals

The goal of this project is to study the dynamics and control issues involved in thrusterless locomotion, and to demonstrate experimentally that a free-flying multi-arm robot can accurately reposition/reorient itself while pushing off from another body (e.g. a space station) rather than using thrusters. The aim is not to construct a space qualified robot, but rather to demonstrate in two dimensions in the laboratory a design philosophy that will easily extend to three dimensions in space.

More specifically, the goals of this project are to:

- Develop a control paradigm and simple algorithms that aid in thrusterless locomotion.
- Achieve smooth switching between control laws during abrupt changes in kinematic configuration. A challenging aspect of this project is to control a nonlinear plant with changing degrees of freedom.
- Construct an autonomous free-flying multi-arm robot that simulates the zero-g drag-free environment of space.
- Verify the design experimentally.

The Experiment Figure 1.1 depicts the conceptual goal of this research: to move the robot accurately from one place to another without using thrusters. To demonstrate faithfully the drag-free zero-g space environment, a second-generation air cushion vehicle (ACV) was designed and built, with many ideas drawn from previous ACV experiments done at Stanford [1]. These vehicles float on a cushion or air on a very flat granite table. Unlike previous ACV's, our current versions have two SCARA[2] arms to allow cooperative manipulation. A photograph of the experimental hardware is shown in Fig 1.2. At the end of each arm is a small robotic hand that can grasp a cylindrical bar affixed to the side of a granite table. The hand is instrumented with force sensors in its palm and pneumatically

[2]SCARA is an acronym for Selective Compliance Assembly Robot Arm. Such robot arms are designed to be very stiff in the vertical axis while being very compliant or easy to move in the horizontal plane [21].

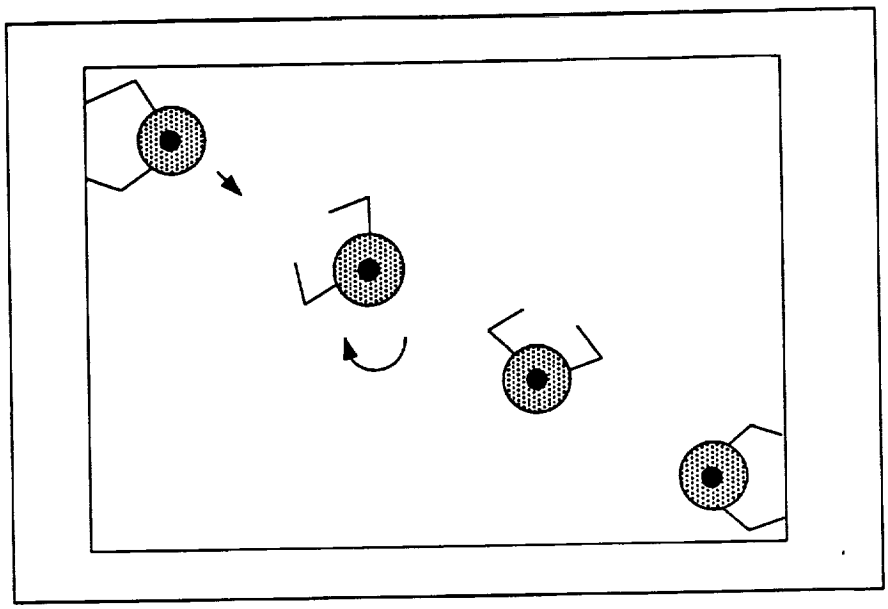

Figure 1.1: **Research Objective**
The robot can push off from one end of the table, rotate, and catch itself at the other end without using thrusters.

actuated finger. There is also a photo emitter/detector pair located between the palm and finger to sense the bar. Located in the center of the vehicle is a momentum wheel to enable turning while in the free-floating configuration. In the free-floating configuration, the robot possesses eight degrees of freedom. With both arms grasping the bar (closed-chain configuration), the number of degrees of freedom drops to four.

1.3 Summary of Results

1.3.1 System Capabilities

The ability for the robot to position itself using only electric motors has been demonstrated. During the three phases of the LEAP maneuver (Push-off, Coast, Docking), a high level strategic controller monitors the system to insure smooth transitions between the phases. While the robot is grasping the bar, the robot can follow trajectories in "momentum" space with a decoupled closed-kinematic-chain controller.

When the bar is released, the robot responds to changes in the configuration and realizes that linear momentum is no longer controllable. Using a combination of bang-bang and

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

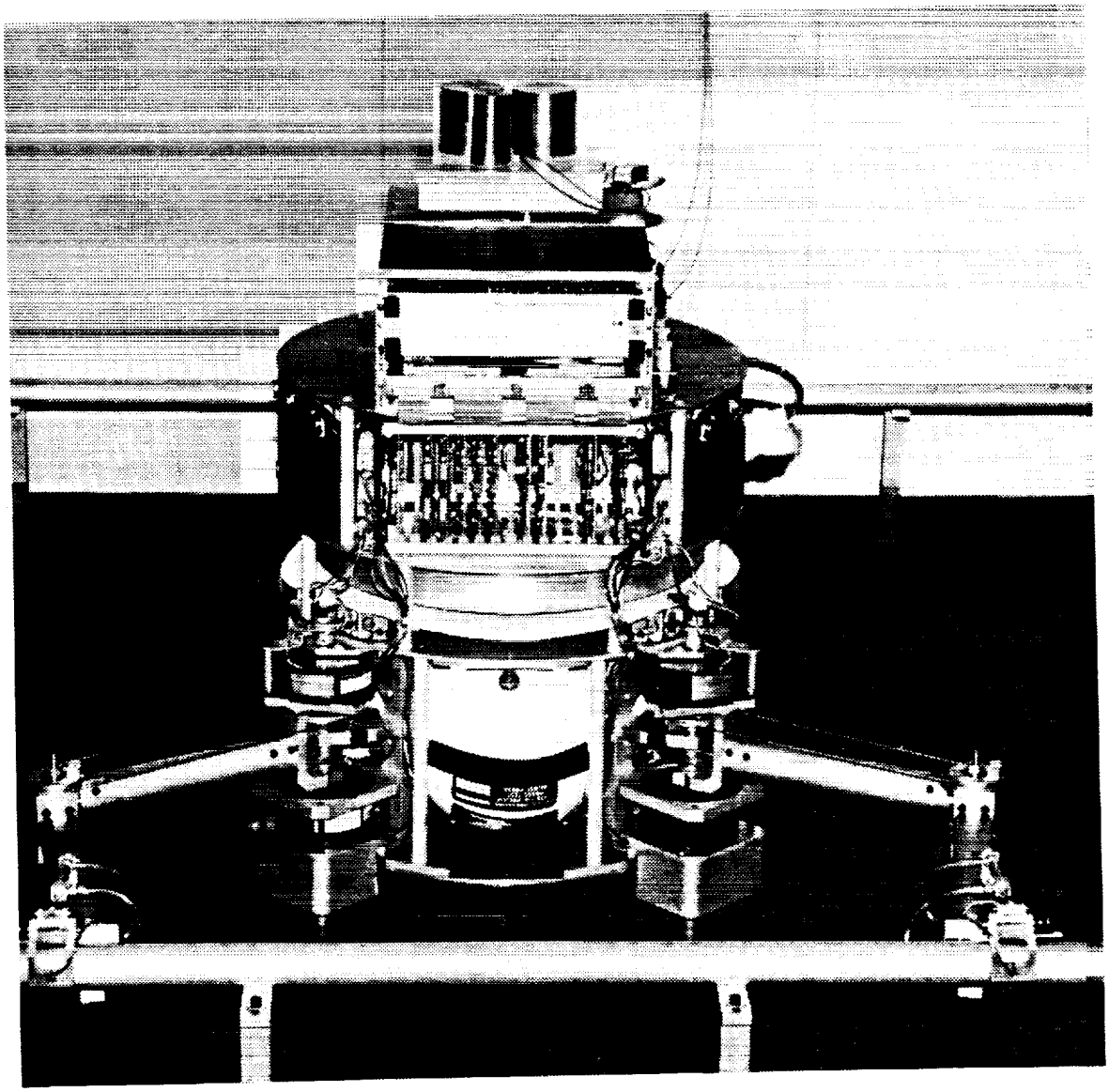

Figure 1.2: **Experimental System**

Overall view of the experimental setup. The two-arm robot floats on an air cushion to simulate the space environment. The vehicle is autonomous, and has onboard power and high-pressure gas. Two robotic hands grasp the bar before and during push-off.

joint PD control, the free-floating robot can orient itself to any desired angle using a momentum wheel.

Finally, when two grasp points are encountered, the control laws are changed yet again to account for the discontinuous change in plant dynamics. During docking, the robot performs a soft-landing by bringing the momentum of the system to zero exponentially.

The operator only commands a desired launch direction and catch angle. All the other state dependent parameters are determined by the strategic controller. The system also possesses onboard error detection logic. Upon determining that an actuator has failed, the strategic controller attempts to put the robot into a safe state.

1.3.2 Contributions

This research makes the following contributions to the field of automatic control, in the particular context of free-flying robots.

- A new approach to using momentum control has been developed. Specifically, this approach, called *system momentum control*, allows the following:
  - Cause a system of rigid bodies to follow a desired momentum trajectory. This allows for precise control of the position and velocity of the mass center for a system of rigid bodies in open-chain or closed-chain configurations.
  - Allow for smooth transitions between open-chain and closed-chain configurations. When a system undergoes abrupt changes in kinematic (holonomic) constraints, the number of degrees of freedom abruptly changes. Since the momentum of the system can be formulated to be independent of the geometry, changes in kinematic constraints do not affect the controller.
  - Formulate the rate of convergence to the trajectory as a linear error equation. For rigid-body systems, the error equations take the form of decoupled second-order systems. The poles of the error equations can be chosen using classical design techniques.
- A control paradigm has been developed to allow switching between different control laws during various phases of a task. This is accomplished by splitting up the controller into a high-level strategic controller and a low-level dynamic controller.

- This control paradigm, implemented experimentally on a free floating space robot, has accomplished the following tasks:
  - Pushing off from a structure in a coordinated two arm maneuver.
  - Rotating the vehicle with the momentum wheel while controlling the configuration of the arms.
  - Landing on another structure.

  Practically, this controller allows the robot to move and orient itself with a minimum amount of propellant. Combining system-momentum control with Cartesian-space control will allow for more general types of tasks such as crawling along a space structure.

- A free-floating space robot has been constructed. This robot is an autonomous vehicle with onboard power, gas thrusters, sensors, and actuators, which floats on an air bearing to simulate in two dimensions the drag-free zero-g environment of space. Versions of this robot will also be used in follow-on research projects.

1.4 Reader's Guide

This dissertation is divided into eight chapters. The following paragraphs summerize each chapter in the thesis.

Chapter One provides motivation for studying thrusterless robotic locomotion, the research goals, and a summary of results. The experiment is briefly described along with a list of contributions.

Chapter Two describes a general design philosophy and overview of the mechanical and electrical design. Following the overview is a more detailed discussion of each sensor and actuator. Specifications on the various sensors are included when deemed appropriate.

Chapter Three includes a derivation of the equations of motion for a two-arm free-flying robot. This chapter can be skipped upon first reading, while referencing the appropriate equations when necessary. It is included here for completeness, and as a reference for further research.

Chapter Four addresses the issues of strategic control, and how higher level task specifications are applied to locomotion. This chapter also includes a brief discussion on the

implications of asynchronous switching of dynamic controllers. Chapter Four concludes with a description of an implementation of a strategic controller.

Chapter Five describes the various control laws used in the experiment. Past work in computed-torque control is discussed, and a new controller, called system momentum control is introduced. Momentum control is derived, followed by a discussion on assumptions and limitations for computed torque.

Chapter Six presents experimental results. A comparison between simulation and experiment is presented to validate the model and design. Finally, experimental and simulation results are presented for an actual complete leap maneuver.

Topics in three dimensional space robotic locomotion are addressed in Chapter Seven. These include an abstract treatment of orientation in three dimensions, controlling attitude in three dimensions, and hardware design issues.

The final chapter, Chapter Eight, summarizes the results. Conclusions drawn from the research are given along with suggestions for future work.

Appendix A details the calibration procedures, and Appendix B derives a simple model for the air bearing.

Chapter 2

Experimental Hardware

This chapter describes the experimental hardware[1] and some of the design decisions that went into its construction. The experimental robot contains many of the subsystems needed for an operational space robot, and faithfully addresses many of the issues in space robotics.

2.1 Design Philosophy

The maxim "Things don't work by accident " and its corollary "Things don't work" realistically describe experimental research. All too often, the transition from simulation to experimental verification fails to achieve meaningful results because of poor judgement and lack of attention to detail. During the design *and* construction of this robot, the successes as well as the failures were studied, so that improvements could be incorporated into future iterations.

One key design decision, made at the beginning of the project, was to make the robot modular, both physically and functionally. Like the layers of a wedding cake, the robot is divided into four sections: high pressure layer, actuator layer, analog/power layer, and a digital layer. Each layer is inclosed in a cylinder with a base radius of 9.5 in. (241.3 mm). The connections between the layers (either gas hoses or electrical wires) are only made through the sides of the cylinders, and not through the bases. Adherence to this rule insured that no layer would interfere with any other layer, and facilitated the inclusion of new capabilities into the design. For example, a year into the design, a 9 inch diameter

[1] Most of the mechanical and electrical designs were done by Marc Ullman and Ross Koningstein, and the author gratefully acknowledges their contributions.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

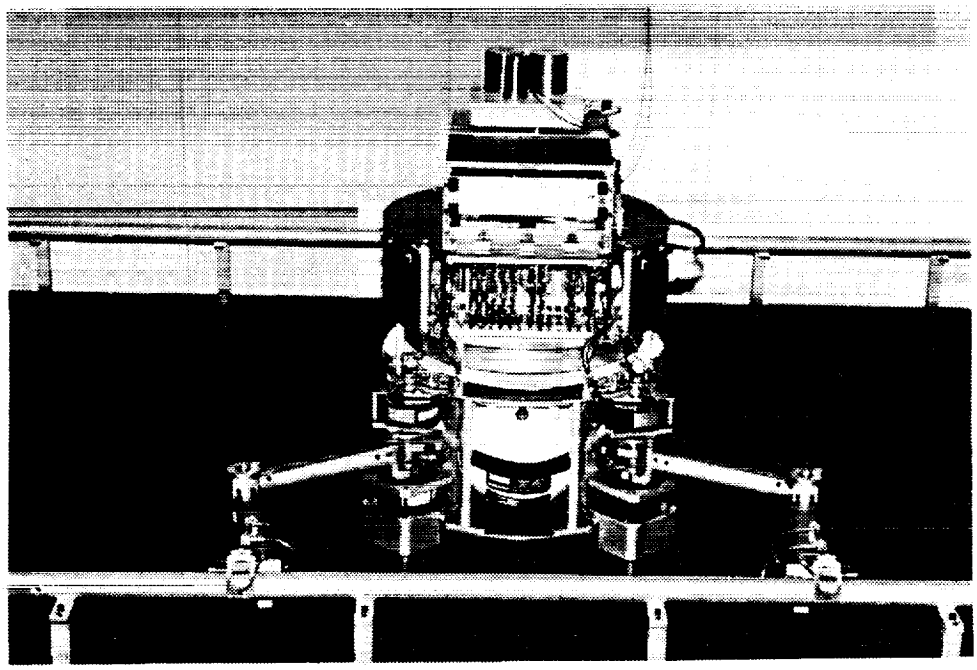

Figure 2.1: **Experimental Free-Flying Space Robot**

The base and arms are capable of planar motion. The robot is only attached to the environment through a thin fiber-optic cable.

momentum wheel was added to the actuator layer without major modification to the rest of the robot.

Another important decision was the use of "off-the-shelf" parts which, when possible, adhered to publically recognized standards. This rule applied to all parts except machined parts and analog electronics, which were designed and built in-house. The reason for this rule is simple: if a part conforms to a public standard, then future parts with enhanced capabilities could be interchanged with the existing one. Also, a public standard usually implies that more than one company is producing the particular item. Thus, the issue of sole source procurement was avoided.

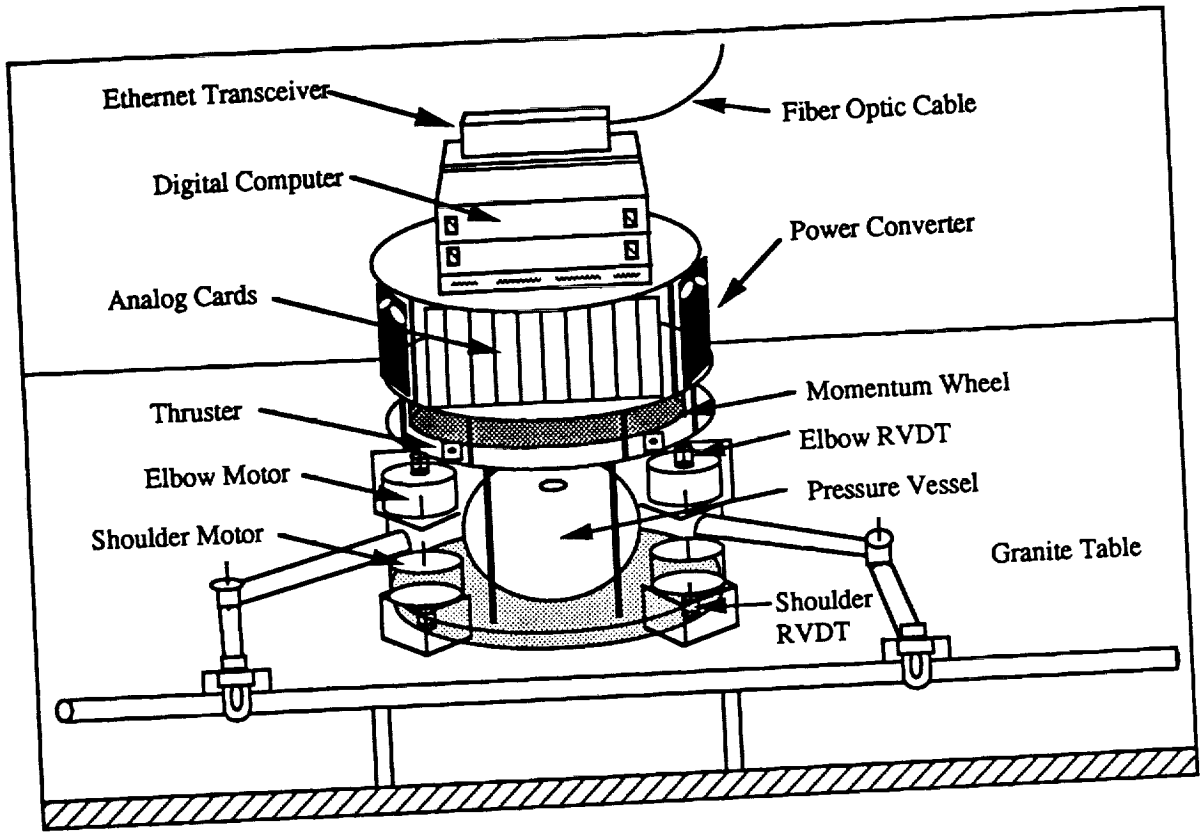

Figure 2.2: **Robot Schematic**

The robot is divided into functional layers: high pressure layer, actuator layer, analog/power layer, and digital electronics layer. The pressure vessels provide compressed air to the robot for floatation, pneumatic actuation, and thrusting. The upper half of the robot contains batteries, analog electronics and the digital computer. Each link is driven by a limited-angle torque motor.

2.2 Mechanical Design

Air Bearing and Support System The robot floats on a air cushion upon a 6 ft × 12 ft (1.83 m × 3.66 m) laboratory grade granite table. The surface of the table has been ground flat to an accuracy of 0.001 in. (25 μm) to provide a level and smooth surface for the maintenance of the air bearing. The air cushion, approximately 50 μm (0.002 in.) thick, requires an air flow of 0.08-0.1 m^3/hr (3.0-3.4 SCFH). The 10 ton table is kinematically

supported on three leveling wedges. Barring seismic disturbances[2], the table can be level to ± 42 μrad. Affixed to the sides of the granite table is a bar, consisting of vertical columns supporting a 1.5 in. (38.1 mm) diameter horizontal tube. The bar, shown in Figure 2.1, is located 6.5 in. (165.1 mm) above the plane of the table so that the robot can grip and release the bar using pneumatic actuators.

The base plate is constructed of 1 in. thick hexel honeycomb sheet bonded between 1/8 in. (top) and 1/16 in. (bottom) aluminum. This design insures small mass and inertia in the base plate while providing structural rigidity. The bottom plate was lapped flat to 25.4 μm. The air supporting the bearing comes primarily from a center hole, with six additional holes located 208.2 mm from the center. The six secondary holes are constricted with small tubes, and act to provide additional stability and prevent grounding to the bearing when the center of mass of the entire robot is not located directly over the geometric center of the air bearing. Rehsteiner [25] has done an extensive theoretical and experimental study of air bearings. He has shown that off-center forces tend to ground the vehicle or result in accelerating the robot in the direction of maximum flow (the high end of the bearing). To minimize these unwanted effects, the robot's mass center was located near the base geometric center. Similar to Alexander's [1] base plate design, a shallow 101.6 mm $\times$ 0.762 mm plenum was machined under the center hole to provide an initial surface area when initiating the gas flow.

Ideally, one would like a drag-free zero-g air environment in the plane of the table. Rehsteiner [25] provides some formulas to calculate viscous drag and gas flow. For a round flat-bottomed base with outer radius r_o, plenum radius r_i and center of mass located at the geometrical center, the drag force **D** is:

$$\begin{aligned} \mathbf{D} &= -\frac{\mu\pi}{h}(r_o^2 - r_i^2)\,{}^A\boldsymbol{v}^{B^*} \\ \frac{\mu\pi}{h}(r_o^2 - r_i^2) &= 0.0612 \quad [\text{ N-s/m }] \end{aligned} \tag{2.1}$$

where μ is the coefficient of viscosity of the gas (air), h is the thickness of the air bearing (50.8 μm nominal) and ${}^A\boldsymbol{v}^{B^*}$ is the velocity of the center of the base plate with respect to the inertial frame. With a mass of 75 kg, this corresponds to a time constant of approximately

[2]During the Loma Prieta earthquake measuring 7.1 on the Richter scale, the 10 ton granite table was shaken off its primary balance and onto a secondary support structure. The table also managed to embed itself about three inches into a wall. Fortunately, no damage was done to the table or the robot.

20 minutes. The flow rate Q is given by:

$$Q = \frac{mgh^3}{3\mu(r_o^2 - r_i^2)} \tag{2.2}$$

Equation 2.2 clearly shows that the flow rate needed to float the robot increases with the cube of the gap thickness and decreases linearly with the area of the base. The derivation of this equation is given in Appendix B.

Gas and Pressure System Figure 2.3 schematically shows the layout of the pressure system [4]. Air is stored on-board in three spherical pressure vessels (maximum working pressure of 3500 psi). The tanks are filled via a flexible hose from a gas cylinder. A primary regulator supplies pressures up to 150 psi to the thrusters, while a second regulator provides pressure of 9 psi to the grippers. Air flow to the bearing is controlled through a Ametek Type 7010 flow meter.

The on-board gas capacity is sufficient to float the robot for over six hours, assuming an initial pressure of 1000 psi (6.98MPa). However, if the gas thrusters are used, the gas system can sustain the air bearing for only 20 minutes before running out of compressed air. This limitation in the laboratory experiment will also affect the operational capabilities of a space robot. For this reason, it is important to study methods which minimize thruster usage.

2.3 Electrical Design

Unlike fixed base robots, which derive their electricity from the power company, mobile robots must store and condition their own electrical power. This fact severly constrains the usefulness of any mobile robot by limiting time of operation and amount of peak actuator authority. The current design for the free-flying robot allows for about 20 minutes of operation before draining the batteries. On board the robot are two battery packs, each containing two NiCd rechargeable batteries to give raw power at $\pm$ 12 V. The batteries are rated at 7 A-h, and can provide 15 A peak current. The raw power is conditioned by onboard DC-DC power converters, supplying regulated voltages of +5 V @ 10 A, $\pm$12 V @ 2.5 A, and $\pm$15 V @ 2.0 A. When the robot is not in use, and external power supply (15 V @ 15 A) recharges the batteries and supplies electrical power to the robot. The external power cable is removed before operation and experimentation. Of course the external power

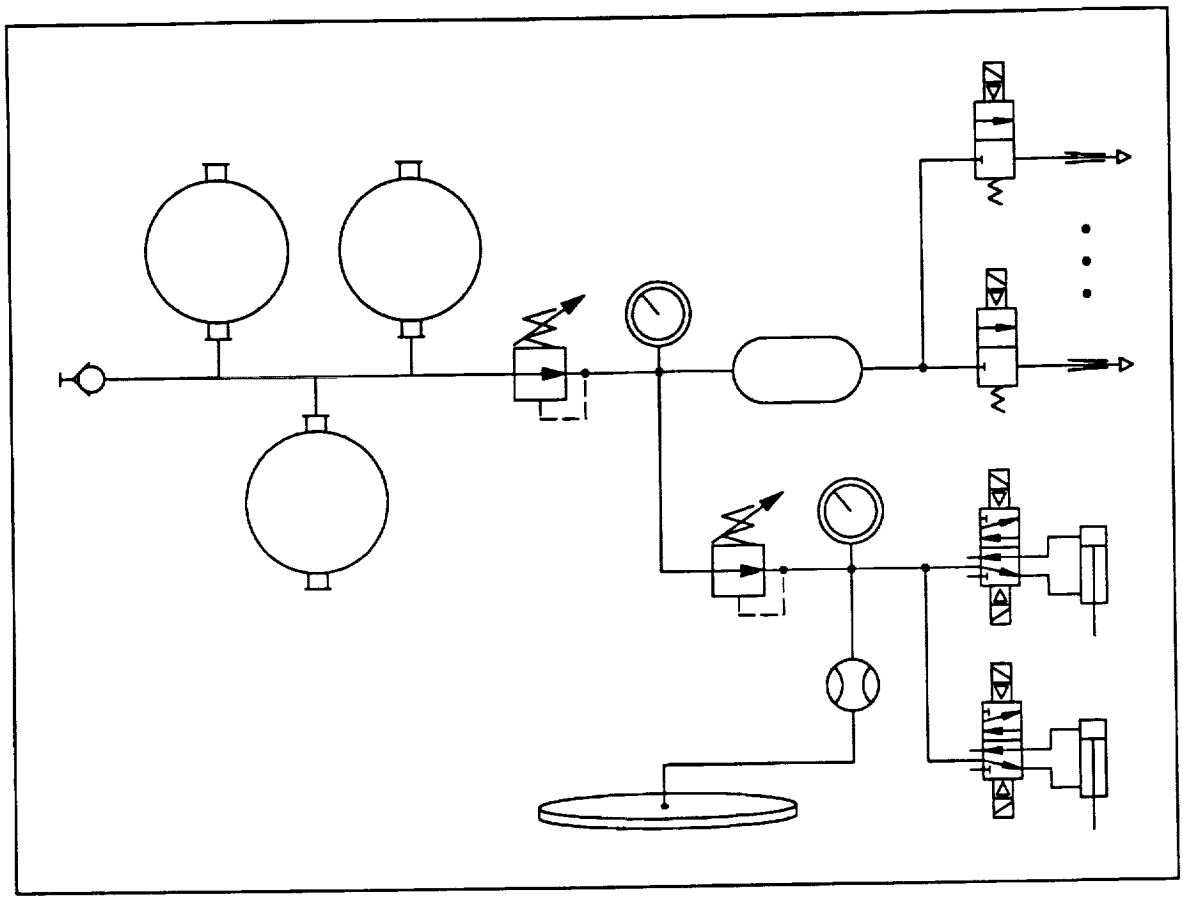

Figure 2.3: **Gas Subsystem Schematic**

Schematic diagram of satellite robot onboard gas subsystem. High pressure air is stored in three pressure vessels. The pressure is regulated to 100 psi for use by gas thrusters. A second regulator reduces the pressure to 10 psi to pneumatically actuate the grippers and supply air for floatation.

cable could be used in place of the batteries, but this limits the range of mobility and the achievable accuracy. Also, when multiple robots cooperate together, the issues of tether fouling will limit their usefulness.

Table 2.1 describes the different analog cards used for switching power, filtering sensor signals, and driving motors.

Number	Component	Comments
1	Power Control Unit	Main power switch and fuses
2	Battery Charging Boards	Charge batteries when connected to external power
2	Safety Board	Drives solenoids and shuts off raw power to bus on error
3	Motor Driver Boards	Each boards drives 2 DC motors
1	RVDT Board	4 rate and position channels per board
1	INS Sensor Board	Filters for accelerometers and rate sensor

Table 2.1: **Analog Cards**

2.4 Sensors

Joint-Angle Sensors Each arm joint is instrumented with a Rotary Variable-Differential Transformer (RVDT) to provide angle measurements. After calibration and filtering, the angles are estimated to an accuracy of ± 2 mrad. The RVDT angle signal is low-pass filtered at 353 Hz. No anti-aliasing filter was employed because the high frequency content of the signal was not sufficient to justify the added phase delay.

An estimate of the angular rate is obtained by passing the position signal through a pseudo-differentiation filter. The transfer function of the filter can be realized as $\frac{-ks}{(as+1)(bs+1)}$, where $1/2\pi a = 12$ Hz , $1/2\pi b = 330$ Hz and $k = 0.968$. At low angular rates ($s < 0.5$ Hz), the filter approximates a pure differentiator. One reason an RVDT sensor was used instead of an encoder was the ease in obtaining reasonable rate information at low frequencies. During normal operation, the maximum angular rate of any joint does not exceed 0.1 Hz. The rate information is accurate to ± 2 mrad/s.

Angular Rate Sensor Mounted on the base plate in an isolation shock mount is the Watson Angular Rate Sensor. This sensor measures the angular velocity of the base plate with respect to the inertial frame (laboratory frame). The sensor consists of two pairs of piezoelectric bender elements, mounted end to end, but rotated 90 degrees as shown in Figure 2.4. The base element is driven at 360 Hz causing it to vibrate. If the instrument is rotated, Coriolis forces cause momentum to be transferred into the perpendicular plane, resulting in a bending of the sense-element.

After calibration, the instrument can detect angular rates of ±1 mrad/s, with average bias uncertainty of ±60 μrad/s over a 5 minute time interval. The maximum angular rate the sensor can detect is 100 deg/s (1.74 rad/s). Typical peak angular rates are 0.6 rad/s for the base body. Table 2.2 gives some important specifications for the instrument.

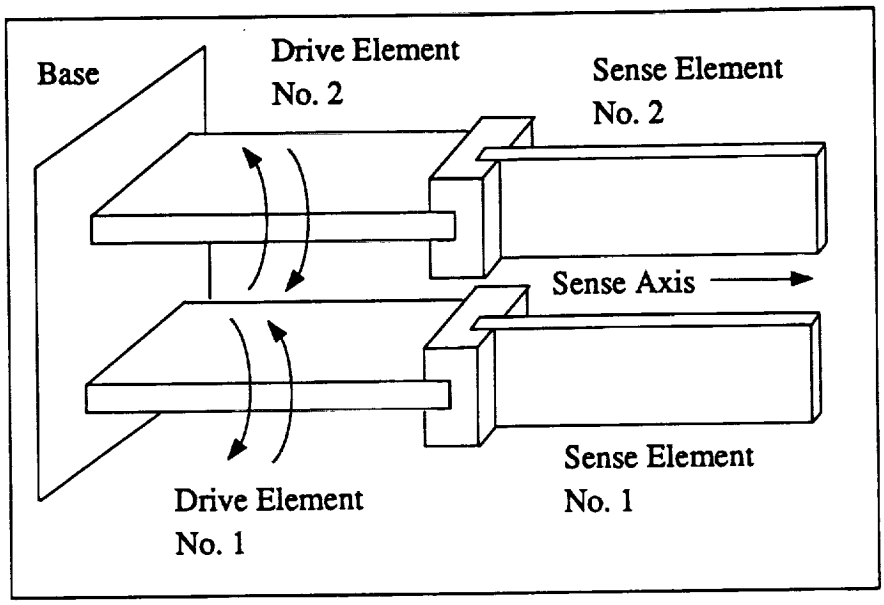

Figure 2.4: **Angular Rate Sensor**

Schematic diagram of the angular rate sensor as a "Tuning Fork" transducer.

Unfortunately, angular rates are not the only source of sense element bendings. External vibrations and accelerations can deflect the sense element, as can acoustical energy which is radiated from the drive element and reflected by its environment to the sense element. In addition, temperature variations causes drift in the DC component of the signal. By using a "tuning fork" configuration, many of these problems are reduced. By driving the two pairs of bending elements at the same frequency but 180 degrees out of phase, a nodal plane created midway between the drive elements cancels radiated energy from the drive elements. Also, external vibration or acceleration cause the sense elements to vibrate in phase, while angular rates cause the sense elements to vibrate out of phase. Thus, differencing the sense element signals achieves common-mode rejection of external vibration and accelerations.

The only other major cause of error is acoustical energy which is radiated from the drive elements, reflected by the environment, and interpreted by the instrument as a rate signal. An electrical analogy is energy being reflected from the end of a transmission line because the impedance at the end is not matched with the terminating resistors. This problem was solved by wrapping the instrument in 1/8 inch Sorbothane, a visco-elastic polymer. The Sorbothane provides acoustical damping and shock absorption by absorbing

Model:	ARS-C131-1A
Power supply:	±15 VDC ±5% 20 mA maximum
Output:	± 10 VDC at full scale angular rate
Sensitivity:	± 100 deg/s full scale
Output current:	± 10 mA maximum
System frequency:	360 Hz nominal
Scale factor error:	2%
Linearity:	< 0.1% full scale
Frequency response:	DC to 30 Hz
Output noise:	5 mV RMS maximum
Life:	50,000 hours MTBF minimum
Shock:	200 g
Weight:	0.110 kg

Table 2.2: **Watson Angular Rate Sensor Specifications**

or transmitting most of the energy.

The angular rate sensor, also known as a "tuning fork gyro", has the advantage over conventional rate gyroscopes of longer life (50,000 hours MTBF) and reduced cost.

Accelerometers The robot is equiped with two Systron Donner 4310 accelerometers. Table 2.3 lists the major attributes of the instrument. Historically, accelerometers have not been used in robotics because of low signal-to-noise ratio in the presence of gravity. It was hoped that the air bearing would afford a stable environment so that accelerations down to the 100 μg level could be measured to give reasonable velocity and position measurements. Unfortunately, drift in the instrument and environmental disturbances did not allow for centimeter accuracy in position over a one minute time scale. However, a scheme that shows good promise in the future is to use a Kalman filter with position measurements from off-board vision combined with accelerometer data to give smooth estimates of position and velocity.

LED Sensor/detector The LED's are used to detect objects which come into close proximity to the grippers. They are driven by very short pulses of current (100 mA) at a nominal period of 1.4 kHz. During one duty cycle, the LED is on for 12 μs and off for 700 μs. The detector, a phototransistor, senses the short pulse and extends them in order to provide a constant DC level. Pulse extension is accomplished by a retriggerable monostable multivibrator (74LS122), with a period of 2 ms. Therefore, any break in the

Model:	Systron Donner 4310
Power supply:	±15 VDC ±10% at 10 mA maximum
Output:	±7.5 VDC at full range
Sensitivity:	±1 g full scale
Output current:	±3 mA maximum
Zero Output (Null):	< 0.05% full range
Linearity:	< 0.05% full scale
Natural frequency:	50 - 250 Hz
Output noise:	< 7.5 mV RMS
Resolution	< 0.001% full range
Shock:	100g for 11 ms
Weight:	128 grams

Table 2.3: **Linear Servo Accelerometer**

light pulses of longer than 2 ms will cause the output of the detector to drop from 5 volts to 0 volts.

Force Sensors The robot is equiped with three force sensors embedded into each gripper. Two sensors are located in the palm, and the other sensor is located in the pneumatically actuated finger. When calibrated, the sensor can detect forces between 0 N and 9 N to an accuracy of 1% of full scale.

Recent advances in micro machining and surface mounted devices (SMD) technology has given rise to solid-state force sensors, pressure sensors, and accelerometers. For example, a small force sensor (3 mm x 3 mm) can be produced by micro machining a thin shelf in the silicon and growing a half bridge circuit on it. Changes in applied force produce linear changes in resistance. The advantage this method has over strain gages is that the force sensor is located at the point of interest, and not some distance away. Strain gages are usually attached to a beam element, and infer the force at the tip by measuring the strain or bending in the beam element. To achieve greater sensitivity, a longer beam is necessary which introduces higher order dynamics. It is also more difficult to measure applied forces on a smooth surface using an array of strain gages. These problems can be overcome by using an array of solid-state force sensors.

2.5 Actuators

There are two basic types of actuators on the robot: electric DC motors and pneumatically actuated solenoids. The five motors supply torque to the four joints in the arms and to the momentum wheel. Eight thrusters, mounted in 4 orthogonal pairs on a flat lexan thruster plate can also supply additional momentum to the vehicle. During normal operation (no thrusting), the air supply can float the robot for about six hours if the tanks are initially pressurized to 1000 psi (6.98 MPa). If the thrusters are used, however, the vehicle can only float for 15 to 20 minutes before grounding out.

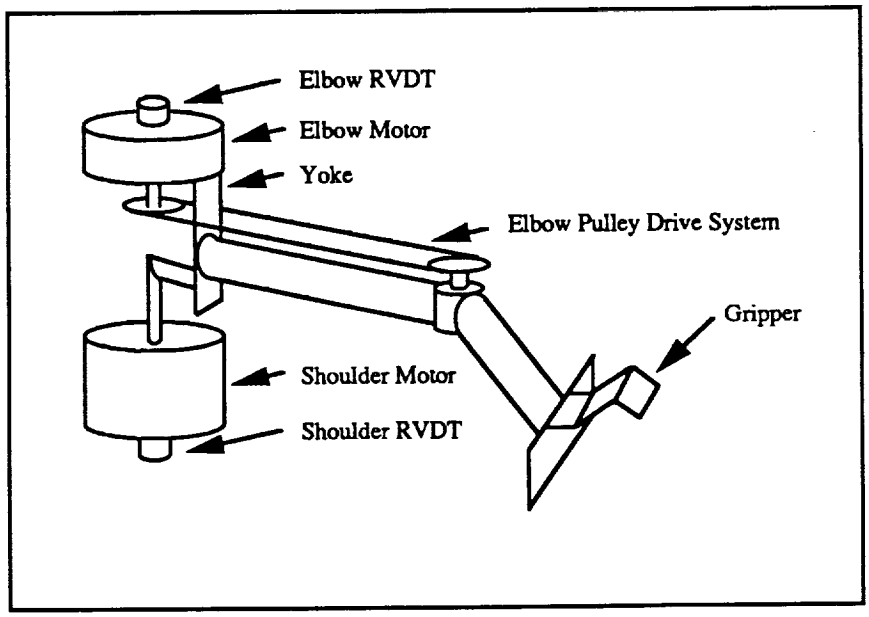

Figure 2.5: **Arm Schematic**

Each link is driven by a limited-angle torque motor; the arm is equipped with joint angle and optical endpoint sensors.

Motors Each link is driven by a limited-angle DC torque motor. These motors were chosen for their nearly frictionless operations. Each arm has two motors, one for the shoulder and one for the elbow. Figure 2.5 shows a schematic of the arms and motors. The shoulder joint is direct drive, while the the elbow joint is driven through a cable and pulley system. The pulley is geared 4:5 to extend the range of the elbow joint at the expense of reduced torque capability. This configuration allows the elbow motors to be mounted on

the shoulder, and drastically reduces the effective inertia of the upper-arm link. The elbow motor is carried by a yoke attached to the shoulder-motor shaft to allow it to operate in its full range independently of the shoulder position.

The robot is also equipped with a DC brush servo motor to drive the momentum wheel. This motor was chosen so as to supply enough torque to rotate the robot 180 deg in 10 s while the robot is in a nominal position [3]. Due to onboard power constraints, only three of the five motors can be run at maximum torque at the same time.

Base Parameters	
Diameter	0.4826 m
Mass	67.96 kg
Inertia	3.29 kg-m^2
Shoulder Link Parameters	
Length	0.3048 m
Mass	1.9231 kg
Inertia	0.0238 kg-m^2
Hub to Link Center of Mass[a]	59.4 mm
Motor Peak Torque	0.91 N-m
Elbow Link Parameters	
Length	0.3015 m
Mass	0.3382 kg
Inertia	0.00416 kg-m^2
Hub to Center of Mass	0.1058 m
Motor Peak Torque	0.59 N-m
Momentum Wheel Parameters	
Mass	3.2943 kg
Inertia	0.1025 kg-m^2
Motor Peak Torque	0.54 N-m

[a]The center of mass is offset by 3.7 mm from the link centerline.

Table 2.4: **Arm and Base Specifications Table**

Grippers Each arm is equipped with a special gripper to allow the robot to detect and grasp a bar surrounding the granite table. The reason for the left and right handedness[4] in the design, apart from preserving symmetry, was to allow a wider range of grips. Each hand has one pneumatically actuated finger and solid-state force sensors located in the

[3]The inertia of the robot taken about the system mass center is a function of configuration.

[4]Each gripper is the mirror image of the other.

palm and finger. There is also a photo emitter/detector pair to sense when an object is within its grasp. The center of mass of each hand is located at the wrist, which is free to pivot along an axis perpendicular to the table. The palm and finger are padded with Sorbothane to supply compliance and prevent bouncing. Although there is no motor in the wrist, the air tubes and wires running through the hand supply a small restoring force to bring the hands to a nominal position. This force was considered small and was not modelled. Figure 2.6 shows a picture of the gripper.

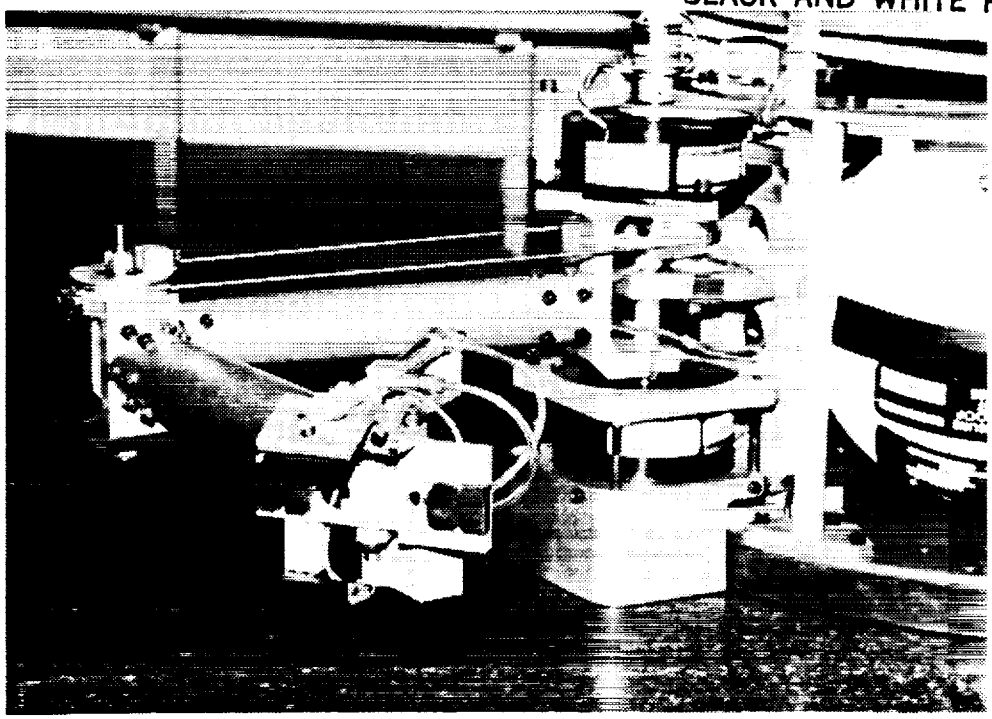

Figure 2.6: **Gripper**

Close up picture of the gripper. Each hand is instrumented with three forces sensors and an optical sensor to sense the bar. The finger is pneumatically actuated.

2.6 Computer System

The computer system incorporates the UNIX development environment with high-performance real-time hardware located onboard the robot. A heterogeneous network of Sun Workstations, running the UNIX operating system, allows a window-based environment for programming, debugging, analysis and simulation. The real-time computer communicates

Number	Component	Manufacturer & Model	Comments
1	Processor	Motorola MVME-147	68030 / 68882 with 4M RAM
1	Analog Input	Xycom XVME-590/3	16 Channel Analog Input
1	Analog Output	Acromag AVME-9210	8 Channel Analog Output
1	Digital I/O	Xycom XVME-290	32 Channel Digital I/O with Timer
1	Ethernet Transceiver	Versitron LE-220	Fiber Optic Ethernet Transceiver

Table 2.5: **Real-Time Computer Components**

with the network via a fiber-optic ethernet cable (see Figure 2.2), to allow programs to be downloaded and experimental data uploaded to the server's hard disk. The computer

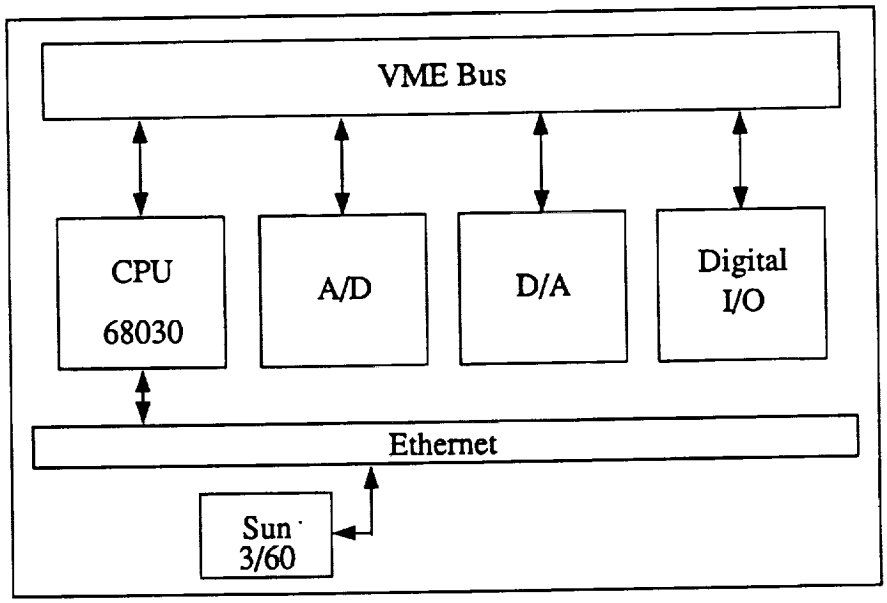

Figure 2.7: **Hardware Architecture**

The computer system combines the Sun UNIX development environment with a high-performance real-time computer.

layout is shown schematically in Figure 2.7. All Sun Workstations in the lab are connected via ethernet and gateways to a local file server. The real-time computer is also connected to the network and looks to the other machines as another diskless workstation. The main CPU card, a Motorola MVME 147 board consists of 4M RAM, a 68030/68882 processor, 4 serial ports and an ethernet port. All the analog electronics interface to the digital electronics through a digital transition module, which connects sensor signals to the digital board's P2 connectors. Table 2.5 summarizes the real-time system hardware.

Chapter 3

Derivations of the Equations of Motion for LEAP Vehicle

3.1 Introduction

This chapter describes a general method for deriving the equations of motion for a free-flying robot. Although alternate formulations have been done for one-arm [1] and two-arm [5] robots, this derivation will focus upon extracting momentum equations. These equations, presented at the end of this chapter as Equation 3.29, will be used in Chapter 5 in conjunction with computed-torque control.

Figure 3.1 shows a schematic of a generic two-arm robot with a momentum wheel. In the free-flying configuration, the robot has 8 DOF. Of course other configurations are possible, namely, whenever one or two arms grasp another structure. When both arms grab a bar, the system is said to be in a closed-chain configuration. In the closed-chain configuration, the number of degrees of freedom drop to 4, and the equations of motion can be made to look very different. By formulating the equations of motion in terms of momentum as in Equation 3.29, only one set of equations is needed, because momentum is a dynamic quantity that is independent of the geometry. The kinematic constraint will be accounted for through the nonholonomic constraint equations of section 3.3.1

3.2 Definitions of the Generalized Speeds

Kane's method [15] will be used for deriving the equations of motion for the LEAP system. Both the free-flying and closed-kinematic chain configuration (arms grabbing the bar) will be derived. This process involves choosing a set of generalized coordinates and generalized speeds, finding the partial velocities, and then deriving the dynamical equations of motion. Having done this, the closed-kinematic-chain derivation is a special case of the open-chain derivation. As a cross-check to the derivation, a computer generated version of the equations of motion were also performed using the software program SD/EXACT [31]. This program produces as its output a FORTRAN subroutine containing the equations of motion. The results of the SD/EXACT program were simplified using MACSYMA [2], a symbolic manipulation package. At the present time, SD/EXACT does not offer the flexibility in choosing generalized velocities and coordinates, which can give equivalent but simpler equations of motion. For this reason, most of the analysis was performed by hand.

Although not unique, the choice of generalized coordinates is predicated on ease of computation. The generalized coordinates represent quantities that are easily measured, as shown in Figure 3.1. The only issue that remains after choosing a set of generalized coordinates q_i is to determine the zero or reference configuration. Three criteria were used to pick the unique reference configuration.

1. The "x" direction of the base should point towards the front of the vehicle to conform with conventional navigation nomenclature. In deriving equations of motions for an airplane, the "x" direction points in the forward direction.

2. The "x" direction of each link should point along the length of the arm. In beam theory, the body coordinates are defined such that the "x" direction is along the beam and the "y" direction is along the direction of deflection.

3. In the reference configuration when all the q_i's are zero, all the body coordinates should line up. For example, $\boldsymbol{b}_1$ and $\boldsymbol{f}_1$ would be parallel.

Note that all subscripts begin at zero instead of one (i.e. the first element of the vector $\boldsymbol{u}$ is u_0). This corresponds to the definition of vector elements in the C language, where the first element of the array x is x[0]. Also, the terms $\sin(q_i)$ and $\cos(q_i)$ are abbreviated as s_i and c_i.

The choice of generalized speeds is also not unique: however, they must be affine in the derivatives of the generalized coordinates. In the *Lagrangian* formulation of the equations

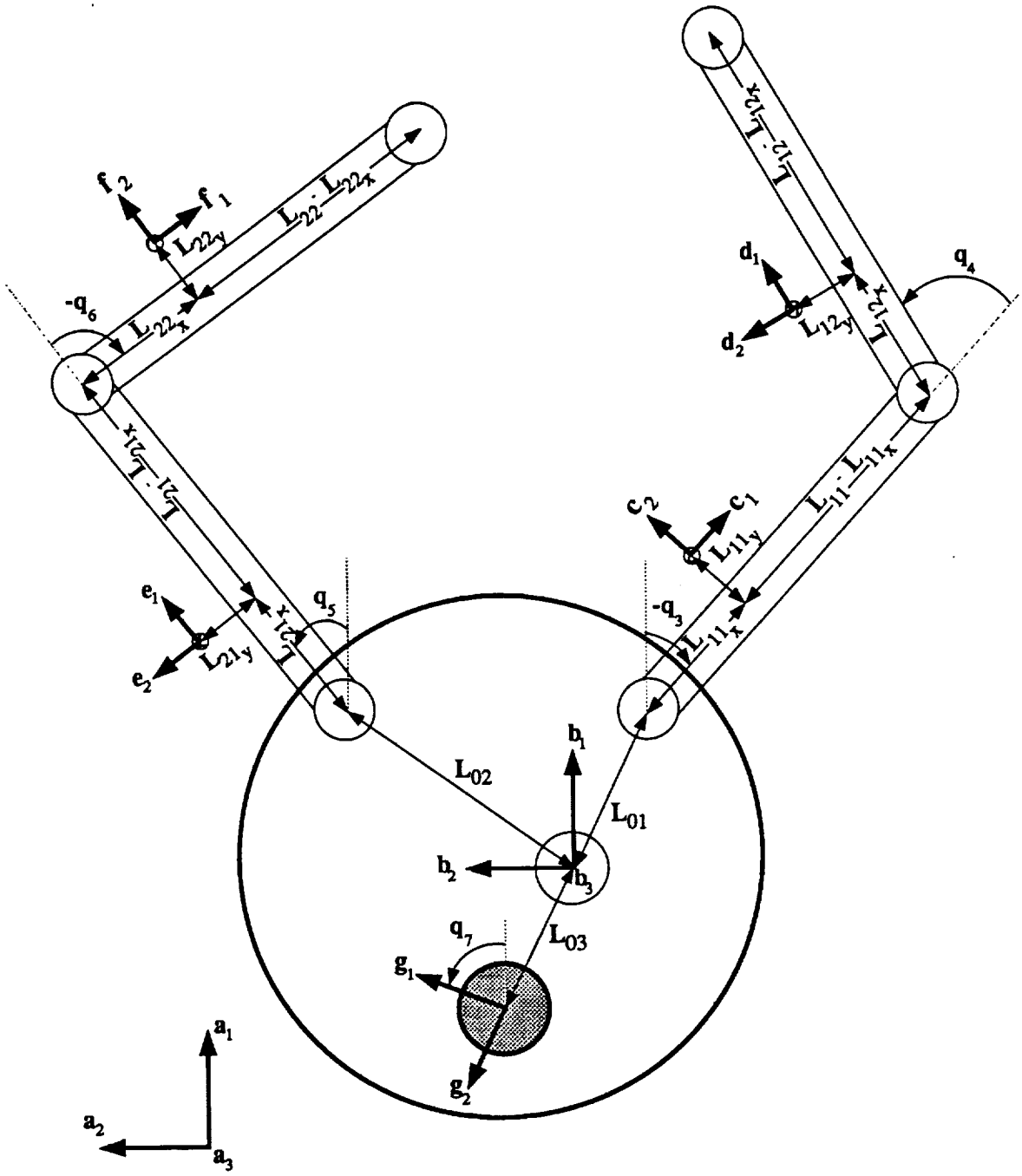

Figure 3.1: **Dynamic Model of the Mobile Space Robot**

Model description is in the text.

of motion, the identity map relates the derivatives of the generalized coordinates to the generalized speeds. This apparent simplification can lead to complicated expressions in the dynamical equations of motion. A choice of generalized speeds which has been found to give simple expressions for the equations-of-motion are:

$$\begin{aligned}
u_0 &\triangleq {}^A\boldsymbol{v}^{B^*} \cdot \boldsymbol{b}_1 = \dot{q}_0 \cos(q_2) + \dot{q}_1 \sin(q_2) \\
u_1 &\triangleq {}^A\boldsymbol{v}^{B^*} \cdot \boldsymbol{b}_2 = \dot{q}_1 \cos(q_2) - \dot{q}_0 \sin(q_2) \\
u_2 &\triangleq {}^A\boldsymbol{\omega}^{B} \cdot \boldsymbol{b}_3 = \dot{q}_2 \\
u_3 &\triangleq {}^A\boldsymbol{\omega}^{C} \cdot \boldsymbol{b}_3 = \dot{q}_2 + \dot{q}_3 \\
u_4 &\triangleq {}^A\boldsymbol{\omega}^{D} \cdot \boldsymbol{b}_3 = \dot{q}_2 + \dot{q}_3 + \dot{q}_4 \\
u_5 &\triangleq {}^A\boldsymbol{\omega}^{E} \cdot \boldsymbol{b}_3 = \dot{q}_2 + \dot{q}_5 \\
u_6 &\triangleq {}^A\boldsymbol{\omega}^{F} \cdot \boldsymbol{b}_3 = \dot{q}_2 + \dot{q}_5 + \dot{q}_6 \\
u_7 &\triangleq {}^A\boldsymbol{\omega}^{G} \cdot \boldsymbol{b}_3 = \dot{q}_2 + \dot{q}_7
\end{aligned} \tag{3.1}$$

where the velocity of point B^* in reference frame A is noted by ${}^A\boldsymbol{v}^{B^*}$. Using these definitions for the generalized speeds, the velocities of the points of interest become:

$$\begin{aligned}
{}^A\boldsymbol{v}^{B^*} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 \\
{}^A\boldsymbol{v}^{P_3} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{01}\,\boldsymbol{r}_2 \\
{}^A\boldsymbol{v}^{C^*} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{01}\,\boldsymbol{r}_2 + u_3\,L_{11_x}\,\boldsymbol{c}_2 - u_3\,L_{11_y}\,\boldsymbol{c}_1 \\
{}^A\boldsymbol{v}^{P_4} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{01}\,\boldsymbol{r}_2 + u_3\,L_{11}\,\boldsymbol{c}_2 \\
{}^A\boldsymbol{v}^{D^*} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{01}\,\boldsymbol{r}_2 + u_3\,L_{11}\,\boldsymbol{c}_2 + u_4\,L_{12_x}\,\boldsymbol{d}_2 - u_4\,L_{12_y}\,\boldsymbol{d}_1 \\
{}^A\boldsymbol{v}^{P_5} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{01}\,\boldsymbol{r}_2 + u_3\,L_{11}\,\boldsymbol{c}_2 + u_4\,L_{12}\,\boldsymbol{d}_2
\end{aligned} \tag{3.2}$$

$$\begin{aligned}
{}^A\boldsymbol{v}^{P_6} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{02}\,\boldsymbol{r}_4 \\
{}^A\boldsymbol{v}^{E^*} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{02}\,\boldsymbol{r}_4 + u_5\,L_{21_x}\,\boldsymbol{e}_2 - u_5\,L_{21_y}\,\boldsymbol{e}_1 \\
{}^A\boldsymbol{v}^{P_7} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{02}\,\boldsymbol{r}_4 + u_5\,L_{21}\,\boldsymbol{e}_2 \\
{}^A\boldsymbol{v}^{F^*} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{02}\,\boldsymbol{r}_4 + u_5\,L_{21}\,\boldsymbol{e}_2 + u_6\,L_{22_x}\,\boldsymbol{f}_2 - u_6\,L_{22_y}\,\boldsymbol{f}_1 \\
{}^A\boldsymbol{v}^{P_8} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{02}\,\boldsymbol{r}_4 + u_5\,L_{21}\,\boldsymbol{e}_2 + u_6\,L_{22}\,\boldsymbol{f}_2 \\
{}^A\boldsymbol{v}^{G^*} &= u_0\,\boldsymbol{b}_1 + u_1\,\boldsymbol{b}_2 + u_2\,L_{03}\,\boldsymbol{r}_6
\end{aligned} \tag{3.3}$$

3.2.1 Accelerations

By differentiating the above velocities with respect to time in the inertial frame A, the following eight accelerations of interest are found:

$$
\begin{aligned}
{}^A\boldsymbol{a}^{B^*} &= \dot{u}_0\,\boldsymbol{b}_1 + \dot{u}_1\,\boldsymbol{b}_2 - u_1u_2\,\boldsymbol{b}_1 + u_0u_2\,\boldsymbol{b}_2 \\
{}^A\boldsymbol{a}^{C^*} &= \dot{u}_0\,\boldsymbol{b}_1 + \dot{u}_1\,\boldsymbol{b}_2 - u_1u_2\,\boldsymbol{b}_1 + u_0u_2\,\boldsymbol{b}_2 + \dot{u}_2L_{01}\,\boldsymbol{r}_2 + \dot{u}_3L_{11_x}\,\boldsymbol{c}_2 - \\
&\quad \dot{u}_3L_{11_y}\,\boldsymbol{c}_1 - u_2{}^2L_{01}\,\boldsymbol{r}_1 - u_3{}^2L_{11_x}\,\boldsymbol{c}_1 - u_3{}^2L_{11_y}\,\boldsymbol{c}_2 \\
{}^A\boldsymbol{a}^{D^*} &= \dot{u}_0\,\boldsymbol{b}_1 + \dot{u}_1\,\boldsymbol{b}_2 - u_1u_2\,\boldsymbol{b}_1 + u_0u_2\,\boldsymbol{b}_2 + \dot{u}_2L_{01}\,\boldsymbol{r}_2 + \dot{u}_3L_{11}\,\boldsymbol{c}_2 + \\
&\quad \dot{u}_4L_{12_x}\,\boldsymbol{d}_2 - \dot{u}_4L_{12_y}\,\boldsymbol{d}_1 - u_2{}^2L_{01}\,\boldsymbol{r}_1 - u_3{}^2L_{11}\,\boldsymbol{c}_1 - u_4{}^2L_{12_x}\,\boldsymbol{d}_1 - u_4{}^2L_{12_y}\,\boldsymbol{d}_2 \\
{}^A\boldsymbol{a}^{E^*} &= \dot{u}_0\,\boldsymbol{b}_1 + \dot{u}_1\,\boldsymbol{b}_2 - u_1u_2\,\boldsymbol{b}_1 + u_0u_2\,\boldsymbol{b}_2 + \dot{u}_2L_{02}\,\boldsymbol{r}_4 + \dot{u}_5L_{21_x}\,\boldsymbol{e}_2 - \\
&\quad \dot{u}_5L_{21_y}\,\boldsymbol{e}_1 - u_2{}^2L_{02}\,\boldsymbol{r}_3 - u_5{}^2L_{21_x}\,\boldsymbol{e}_1 - u_5{}^2L_{21_y}\,\boldsymbol{e}_2 \\
{}^A\boldsymbol{a}^{F^*} &= \dot{u}_0\,\boldsymbol{b}_1 + \dot{u}_1\,\boldsymbol{b}_2 - u_1u_2\,\boldsymbol{b}_1 + u_0u_2\,\boldsymbol{b}_2 + \dot{u}_2L_{02}\,\boldsymbol{r}_4 + \dot{u}_5L_{21}\,\boldsymbol{e}_2 + \\
&\quad \dot{u}_6L_{22_x}\,\boldsymbol{f}_2 - \dot{u}_6L_{22_y}\,\boldsymbol{f}_1 - u_2{}^2L_{02}\,\boldsymbol{r}_3 - u_5{}^2L_{21}\,\boldsymbol{e}_1 - u_6{}^2L_{22_x}\,\boldsymbol{f}_1 - u_6{}^2L_{22_y}\,\boldsymbol{f}_2 \\
{}^A\boldsymbol{a}^{G^*} &= \dot{u}_0\,\boldsymbol{b}_1 + \dot{u}_1\,\boldsymbol{b}_2 - u_1u_2\,\boldsymbol{b}_1 + u_0u_2\,\boldsymbol{b}_2 + \dot{u}_2L_{03}\,\boldsymbol{r}_6 - u_2{}^2L_{03}\,\boldsymbol{r}_5 \qquad (3.4) \\
{}^A\boldsymbol{a}^{P_5} &= \dot{u}_0\,\boldsymbol{b}_1 + \dot{u}_1\,\boldsymbol{b}_2 - u_1u_2\,\boldsymbol{b}_1 + u_0u_2\,\boldsymbol{b}_2 + \dot{u}_2L_{01}\,\boldsymbol{r}_2 + \dot{u}_3L_{11}\,\boldsymbol{c}_2 + \\
&\quad \dot{u}_4L_{12}\,\boldsymbol{d}_2 - u_2{}^2L_{01}\,\boldsymbol{r}_1 - u_3{}^2L_{11}\,\boldsymbol{c}_1 - u_4{}^2L_{12}\,\boldsymbol{d}_1 \\
{}^A\boldsymbol{a}^{P_8} &= \dot{u}_0\,\boldsymbol{b}_1 + \dot{u}_1\,\boldsymbol{b}_2 - u_1u_2\,\boldsymbol{b}_1 + u_0u_2\,\boldsymbol{b}_2 + \dot{u}_2L_{02}\,\boldsymbol{r}_4 + \dot{u}_5L_{21}\,\boldsymbol{e}_2 + \\
&\quad \dot{u}_6L_{22}\,\boldsymbol{f}_2 - u_2{}^2L_{02}\,\boldsymbol{r}_3 - u_5{}^2L_{21}\,\boldsymbol{e}_1 - u_6{}^2L_{22}\,\boldsymbol{f}_1
\end{aligned}
$$

3.2.2 Partial Velocities and Partial Angular Velocities

The next step in the derivation of the equations of motion is to calculate the partial velocities ${}^A\boldsymbol{v}_r^{P_i}$ and partial angular velocities ${}^A\boldsymbol{\omega}_r^B$ where

$$ {}^A\boldsymbol{v}_r^{P_i} \triangleq \frac{\partial\, {}^A\boldsymbol{v}^{P_i}}{\partial u_r} \tag{3.5} $$

$$ {}^A\boldsymbol{\omega}_r^B \triangleq \frac{\partial\, {}^A\boldsymbol{\omega}^B}{\partial u_r} \tag{3.6} $$

The tables of the partial velocities of the points of interest and the partial angular velocities of the bodies of interest for the robot are given on the following page.

3.3 Equations of Motion

The equations of motion can now be derived using *Kane's dynamical equations* [15] which state that:

$$ F_r + F_r^* = 0 \qquad r = 1 \ldots n $$

where n is the number of generalized coordinates or degrees of freedom, and

$$ F_r^* = \sum_{i=0}^{\nu} -m_i\, {}^A\boldsymbol{v}_r^{P_i} \cdot \boldsymbol{a}_i $$

$$ F_r = \sum_{i=0}^{\nu} {}^A\boldsymbol{v}_r^{P_i} \cdot \boldsymbol{R}_i $$

ν is the total number of particles and $\boldsymbol{R}_i$ is the resultant of all forces acting on particle P_i.

By combining terms, one can put the equations of motion into the form

$$ \begin{aligned} \boldsymbol{M}\,\dot{\boldsymbol{u}} - \boldsymbol{c} &= \boldsymbol{T}\boldsymbol{\tau} + \boldsymbol{f}_{external} \\ \dot{\boldsymbol{q}} &= \boldsymbol{Y}^{-1}\boldsymbol{u} \end{aligned} \tag{3.7} $$

where $\boldsymbol{M}$ is a symmetric positive definite mass matrix, $\boldsymbol{c}$ is an 8×1 vector of Coriolis and centrifugal terms, $\boldsymbol{\tau}$ is a 5×1 vector of applied motor torques and $\boldsymbol{f}_{external}$ is an 8×1 vector of all other external unmodelled active forces acting on the system. We will assume that $\boldsymbol{f}_{external} = 0$ in the following derivation. The *unimodular* matrix $\boldsymbol{Y}^{-1}$ defines the kinematic relationship between $\dot{\boldsymbol{q}}$ and $\boldsymbol{u}$.

r	${}^A\boldsymbol{v}_r^{B^*}$	${}^A\boldsymbol{v}_r^{P_3}$	${}^A\boldsymbol{v}_r^{C^*}$	${}^A\boldsymbol{v}_r^{P_4}$
0	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$
1	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$
2	0	$L_{01}\,\boldsymbol{r}_2$	$L_{01}\,\boldsymbol{r}_2$	$L_{01}\,\boldsymbol{r}_2$
3	0	0	$L_{11_x}\,\boldsymbol{c}_2 - L_{11_y}\,\boldsymbol{c}_1$	$L_{11}\,\boldsymbol{c}_2$
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0

r	${}^A\boldsymbol{v}_r^{D^*}$	${}^A\boldsymbol{v}_r^{P_5}$	${}^A\boldsymbol{v}_r^{P_6}$	${}^A\boldsymbol{v}_r^{E^*}$
0	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$
1	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$
2	$L_{01}\,\boldsymbol{r}_2$	$L_{01}\,\boldsymbol{r}_2$	$L_{02}\,\boldsymbol{r}_4$	$L_{02}\,\boldsymbol{r}_4$
3	$L_{11}\,\boldsymbol{c}_2$	$L_{11}\,\boldsymbol{c}_2$	0	0
4	$L_{12_x}\,\boldsymbol{d}_2 - L_{12_y}\,\boldsymbol{d}_1$	$L_{12}\,\boldsymbol{d}_2$	0	0
5	0	0	0	$L_{21_x}\,\boldsymbol{e}_2 - L_{21_y}\,\boldsymbol{e}_1$
6	0	0	0	0
7	0	0	0	0

r	${}^A\boldsymbol{v}_r^{P_7}$	${}^A\boldsymbol{v}_r^{F^*}$	${}^A\boldsymbol{v}_r^{P_8}$	${}^A\boldsymbol{v}_r^{G^*}$
0	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$	$\boldsymbol{b}_1$
1	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$	$\boldsymbol{b}_2$
2	$L_{02}\,\boldsymbol{r}_4$	$L_{02}\,\boldsymbol{r}_4$	$L_{02}\,\boldsymbol{r}_4$	$L_{03}\,\boldsymbol{r}_6$
3	0	0	0	0
4	0	0	0	0
5	$L_{21}\,\boldsymbol{e}_2$	$L_{21}\,\boldsymbol{e}_2$	$L_{21}\,\boldsymbol{e}_2$	0
6	0	$L_{22_x}\,\boldsymbol{f}_2 - L_{22_y}\,\boldsymbol{f}_1$	$L_{22}\,\boldsymbol{f}_2$	0
7	0	0	0	0

Table 3.1: **The partial velocities**

r	${}^A\boldsymbol{\omega}_r^{B}$	${}^A\boldsymbol{\omega}_r^{C}$	${}^A\boldsymbol{\omega}_r^{D}$	${}^A\boldsymbol{\omega}_r^{E}$	${}^A\boldsymbol{\omega}_r^{F}$	${}^A\boldsymbol{\omega}_r^{G}$
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	$\boldsymbol{b}_3$	0	0	0	0	0
3	0	$\boldsymbol{b}_3$	0	0	0	0
4	0	0	$\boldsymbol{b}_3$	0	0	0
5	0	0	0	$\boldsymbol{b}_3$	0	0
6	0	0	0	0	$\boldsymbol{b}_3$	0
7	0	0	0	0	0	$\boldsymbol{b}_3$

Table 3.2: **The partial angular velocities**

$$M = \begin{bmatrix}
z_1 & 0 & z_2 & z_3 s_3 + z_4 c_3 & z_5 s_{34} + z_6 c_{34} & z_7 s_5 + z_8 c_5 & z_9 s_{56} + z_{10} c_{56} & 0 \\
0 & z_1 & z_{11} & z_4 s_3 - z_3 c_3 & z_6 s_{34} - z_5 c_{34} & z_8 s_5 - z_7 c_5 & z_{10} s_{56} - z_9 c_{56} & 0 \\
z_2 & z_{11} & z_{12} & z_{13} s_3 + z_{14} c_3 & z_{15} s_{34} + z_{16} c_{34} & z_{17} s_5 + z_{18} c_5 & z_{19} s_{56} + z_{20} c_{56} & 0 \\
z_3 s_3 + z_4 c_3 & z_4 s_3 - z_3 c_3 & z_{13} s_3 + z_{14} c_3 & z_{21} & z_{22} s_4 + z_{23} c_4 & 0 & 0 & 0 \\
z_5 s_{34} + z_6 c_{34} & z_6 s_{34} - z_5 c_{34} & z_{15} s_{34} + z_{16} c_{34} & z_{22} s_4 + z_{23} c_4 & z_{24} & 0 & 0 & 0 \\
z_7 s_5 + z_8 c_5 & z_8 s_5 - z_7 c_5 & z_{17} s_5 + z_{18} c_5 & 0 & 0 & z_{25} & z_{26} s_6 + z_{27} c_6 & 0 \\
z_9 s_{56} + z_{10} c_{56} & z_{10} s_{56} - z_9 c_{56} & z_{19} s_{56} + z_{20} c_{56} & 0 & 0 & z_{26} s_6 + z_{27} c_6 & z_{28} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & z_{29}
\end{bmatrix} \tag{3.8}$$

Where the z_i are constants and have the following definitions:

$$\begin{aligned}
z_1 &\triangleq m_0 + m_1 + m_2 + m_3 + m_4 + m_5 \\
z_2 &\triangleq -(m_1 + m_2) L_{01_y} - (m_3 + m_4) L_{02_y} - m_5 L_{03_y} \\
z_3 &\triangleq -m_1 L_{11_x} - m_2 L_{11} \\
z_4 &\triangleq -m_1 L_{11_y} \\
z_5 &\triangleq -m_2 L_{12_x} \\
z_6 &\triangleq -m_2 L_{12_y}
\end{aligned}$$

$$
\begin{aligned}
z_7 &\triangleq -m_3 L_{21_x} - m_4 L_{21} \\
z_8 &\triangleq -m_3 L_{21_y} \\
z_9 &\triangleq -m_4 L_{22_x} \\
z_{10} &\triangleq -m_4 L_{22_y} \\
z_{11} &\triangleq (m_1 + m_2) L_{01_x} + (m_3 + m_4) L_{02_x} + m_5 L_{03_x} \\
z_{12} &\triangleq I_0 + (m_1 + m_2) {L_{01}}^2 + (m_3 + m_4) {L_{02}}^2 + m_5 {L_{03}}^2 \\
z_{13} &\triangleq -m_1 L_{01_x} L_{11_y} + m_1 L_{01_y} L_{11_x} + m_2 L_{01_y} L_{11} \\
z_{14} &\triangleq m_1 L_{01_x} L_{11_x} + m_1 L_{01_y} L_{11_y} + m_2 L_{01_x} L_{11} \\
z_{15} &\triangleq -m_2 L_{01_x} L_{12_y} + m_2 L_{01_y} L_{12_x} \\
z_{16} &\triangleq m_2 L_{01_x} L_{12_x} + m_2 L_{01_y} L_{12_y} \\
z_{17} &\triangleq -m_3 L_{02_x} L_{21_y} + m_3 L_{02_y} L_{21_x} + m_4 L_{02_y} L_{21} \\
z_{18} &\triangleq m_3 L_{02_x} L_{21_x} + m_3 L_{02_y} L_{21_y} + m_4 L_{02_x} L_{21} \\
z_{19} &\triangleq -m_4 L_{02_x} L_{22_y} + m_4 L_{02_y} L_{22_x} \\
z_{20} &\triangleq m_4 L_{02_x} L_{22_x} + m_4 L_{02_y} L_{22_y} \\
z_{21} &\triangleq I_1 + m_1 ({L_{11_x}}^2 + {L_{11_y}}^2) + m_2 {L_{11}}^2 \\
z_{22} &\triangleq -m_2 L_{11} L_{12_y} \\
z_{23} &\triangleq m_2 L_{11} L_{12_x} \\
z_{24} &\triangleq I_2 + m_2 ({L_{12_x}}^2 + {L_{12_y}}^2) \\
z_{25} &\triangleq I_3 + m_3 ({L_{21_x}}^2 + {L_{21_y}}^2) + m_4 {L_{21}}^2 \\
z_{26} &\triangleq -m_4 L_{21} L_{22_y} \\
z_{27} &\triangleq m_4 L_{21} L_{22_x} \\
z_{28} &\triangleq I_4 + m_4 ({L_{22_x}}^2 + {L_{22_y}}^2) \\
z_{29} &\triangleq I_5
\end{aligned}
$$

The choice of generalized speeds determines the complexity of the mass matrix. With the present choice, the mass matrix can be formulated with 28 multiplies and 14 additions. These operations can be performed in parallel, since there this no dependency between the elements of the mass matrix. Also note that the quantities q_1, q_2, and q_3 do not appear in the mass matrix or the nonlinear Coriolis and centrifugal terms.

The nonlinear Coriolis and centrifugal terms are found in the vector $\boldsymbol{c}$ for Equation 3.7, and are functions of q_i, u_i and z_i.

$$
\begin{aligned}
c(0) &= u_1u_2z_1 + {u_2}^2z_{11} + {u_3}^2(z_4s_3 - z_3c_3) + {u_4}^2(z_6s_{34} - z_5c_{34}) + \\
&\quad {u_5}^2(z_8s_5 - z_7c_5) + {u_6}^2(z_{10}s_{56} - z_9c_{56}) \\
c(1) &= -u_0u_2z_1 - {u_2}^2z_2 - {u_3}^2(z_3s_3 + z_4c_3) - {u_4}^2(z_5s_{34} + z_6c_{34}) - \\
&\quad {u_5}^2(z_7s_5 + z_8c_5) - {u_6}^2(z_9s_{56} + z_{10}c_{56}) \\
c(2) &= -u_0u_2z_{11} + u_1u_2z_2 + {u_3}^2(z_{14}s_3 - z_{13}c_3) + {u_4}^2(z_{16}s_{34} - z_{15}c_{34}) + \\
&\quad {u_5}^2(z_{18}s_5 - z_{17}c_5) + {u_6}^2(z_{20}s_{56} - z_{19}c_{56}) \\
c(3) &= -u_0u_2(z_4s_3 - z_3c_3) + u_1u_2(z_3s_3 + z_4c_3) - {u_2}^2(z_{14}s_3 - z_{13}c_3) + \\
&\quad {u_4}^2(z_{23}s_4 - z_{22}c_4) \\
c(4) &= -u_0u_2(z_6s_{34} - z_5c_{34}) + u_1u_2(z_5s_{34} + z_6c_{34}) - {u_2}^2(z_{16}s_{34} - z_{15}c_{34}) - \\
&\quad {u_3}^2(z_{23}s_4 - z_{22}c_4) \\
c(5) &= -u_0u_2(z_8s_5 - z_7c_5) + u_1u_2(z_7s_5 + z_8c_5) - {u_2}^2(z_{18}s_5 - z_{17}c_5) + \\
&\quad {u_6}^2(z_{27}s_6 - z_{26}c_6) \\
c(6) &= -u_0u_2(z_{10}s_{56} - z_9c_{56}) + u_1u_2(z_9s_{56} + z_{10}c_{56}) - {u_2}^2(z_{20}s_{56} - z_{19}c_{56}) - \\
&\quad {u_5}^2(z_{27}s_6 - z_{26}c_6) \\
c(7) &= 0
\end{aligned}
$$

The vector $T\tau$ in Equation 3.7 is given by

$$
T\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \tag{3.9}
$$

Where the τ_i are torques from the five motors. Note that the sum of each element in a column of T is zero. The physical implications of this fact is that no motor can cause a net torque about the system's mass center. The first two rows are zero, signifying that the

actuators have no net effect on the total linear momentum. The Y^{-1} matrix in Equation 3.7 is unimodular and is given by:

$$Y^{-1} = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & 0 & 0 & 0 & 0 & 0 \\ \sin(q_2) & \cos(q_2) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.3.1 Nonholonomic Constraint Equations

When both arms of the robot are grasping the bar, the number of degrees of freedom of the system drop from eight to four. At the tip of each arm, there two nonholonomic or velocity constraints which much be satisfied to insure the closed kinematic-chain configuration. The four constraint equations are:

$$^{A}\boldsymbol{v}^{P_5} \cdot \boldsymbol{b}_1 = 0 \tag{3.10}$$

$$^{A}\boldsymbol{v}^{P_5} \cdot \boldsymbol{b}_2 = 0 \tag{3.11}$$

$$^{A}\boldsymbol{v}^{P_8} \cdot \boldsymbol{b}_1 = 0 \tag{3.12}$$

$$^{A}\boldsymbol{v}^{P_8} \cdot \boldsymbol{b}_2 = 0 \tag{3.13}$$

Substituting Equation 3.2 and Equation 3.3 yields the following set of nonholonomic constraint equations.

$$\boldsymbol{u}_s = \boldsymbol{A}_{sr}\boldsymbol{u}_r \tag{3.14}$$

$$\begin{bmatrix} u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} -\frac{\cos(q_3+q_4)}{L_{11}\sin(q_4)} & -\frac{\sin(q_3+q_4)}{L_{11}\sin(q_4)} & \frac{-L_{01_x}\sin(q_3+q_4)+L_{01_y}\cos(q_3+q_4)}{L_{11}\sin(q_4)} \\ \frac{\cos(q_3)}{L_{12}\sin(q_4)} & \frac{\sin(q_3)}{L_{12}\sin(q_4)} & \frac{L_{01_x}\sin(q_3)-L_{01_y}\cos(q_3)}{L_{12}\sin(q_4)} \\ -\frac{\cos(q_5+q_6)}{L_{21}\sin(q_6)} & -\frac{\sin(q_5+q_6)}{L_{21}\sin(q_6)} & \frac{-L_{02_x}\sin(q_5+q_6)+L_{02_y}\cos(q_5+q_6)}{L_{21}\sin(q_6)} \\ \frac{\cos(q_5)}{L_{22}\sin(q_6)} & \frac{\sin(q_5)}{L_{22}\sin(q_6)} & \frac{L_{02_x}\sin(q_5)-L_{02_y}\cos(q_5)}{L_{22}\sin(q_6)} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix}$$

Differentiating the velocities in Equation 3.2 and Equation 3.3 with respect to time in reference frame A and dotting the acceleration vectors with orthogonal unit vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ gives the following set of equations which expresses $\dot{\boldsymbol{u}}_s$ in terms of $\dot{\boldsymbol{u}}_r, \boldsymbol{u}$ and $\boldsymbol{q}$.

$$\dot{\boldsymbol{u}}_s = \boldsymbol{A}_{sr}\dot{\boldsymbol{u}}_r + \boldsymbol{b}_s \tag{3.15}$$

$$\begin{bmatrix} \dot{u}_3 \\ \dot{u}_4 \\ \dot{u}_5 \\ \dot{u}_6 \end{bmatrix} = \begin{bmatrix} -\frac{\cos(q_3+q_4)}{L_{11}\sin(q_4)} & -\frac{\sin(q_3+q_4)}{L_{11}\sin(q_4)} & \frac{-L_{01_x}\sin(q_3+q_4)+L_{01_y}\cos(q_3+q_4)}{L_{11}\sin(q_4)} \\ \frac{\cos(q_3)}{L_{12}\sin(q_4)} & \frac{\sin(q_3)}{L_{12}\sin(q_4)} & \frac{L_{01_x}\sin(q_3)-L_{01_y}\cos(q_3)}{L_{12}\sin(q_4)} \\ -\frac{\cos(q_5+q_6)}{L_{21}\sin(q_6)} & -\frac{\sin(q_5+q_6)}{L_{21}\sin(q_6)} & \frac{-L_{02_x}\sin(q_5+q_6)+L_{02_y}\cos(q_5+q_6)}{L_{21}\sin(q_6)} \\ \frac{\cos(q_5)}{L_{22}\sin(q_6)} & \frac{\sin(q_5)}{L_{22}\sin(q_6)} & \frac{L_{02_x}\sin(q_5)-L_{02_y}\cos(q_5)}{L_{22}\sin(q_6)} \end{bmatrix} \begin{bmatrix} \dot{u}_0 \\ \dot{u}_1 \\ \dot{u}_2 \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{-u_0u_2\sin(q_3+q_4)+u_1u_2\cos(q_3+q_4)+u_2^2[L_{01_x}\cos(q_3+q_4)+L_{01_y}\sin(q_3+q_4)]+u_3^2L_{11}\cos(q_4)+u_4^2L_{12}}{L_{11}\sin(q_4)} \\ \frac{u_0u_2\sin(q_3)-u_1u_2\cos(q_3)-u_2^2[L_{01_x}\cos(q_3)+L_{01_y}\sin(q_3)]-u_3^2L_{11}-u_4^2L_{12}\cos(q_4)}{L_{12}\sin(q_4)} \\ \frac{-u_0u_2\sin(q_5+q_6)+u_1u_2\cos(q_5+q_6)+u_2^2[L_{02_x}\cos(q_5+q_6)+L_{02_y}\sin(q_5+q_6)]+u_5^2L_{21}\cos(q_6)+u_6^2L_{22}}{L_{21}\sin(q_6)} \\ \frac{u_0u_2\sin(q_5)-u_1u_2\cos(q_5)-u_2^2[L_{02_x}\cos(q_5)+L_{02_y}\sin(q_5)]-u_5^2L_{21}-u_6^2L_{22}\cos(q_6)}{L_{22}\sin(q_6)} \end{bmatrix}$$

With this constraint equation, one can derive the reduced set of the equations of motion by partitioning Equation 3.7 and adjoining the constraint equation as follows

$$\begin{bmatrix} \boldsymbol{M}_{rr} & \boldsymbol{M}_{rs} \\ \boldsymbol{M}_{rs}^T & \boldsymbol{M}_{ss} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_r \\ \dot{\boldsymbol{u}}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_r \\ \boldsymbol{c}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{T}_r \\ \boldsymbol{T}_s \end{bmatrix} \boldsymbol{\tau} \tag{3.16}$$

$$\begin{aligned} \left[\boldsymbol{M}_{rr} + \boldsymbol{M}_{rs}\boldsymbol{A}_{sr} + (\boldsymbol{M}_{rs}\boldsymbol{A}_{sr})^T + \boldsymbol{A}_{sr}^T\boldsymbol{M}_{ss}\boldsymbol{A}_{sr}\right]\dot{\boldsymbol{u}}_r = \\ \left[\boldsymbol{c}_r + \boldsymbol{A}_{sr}^T\boldsymbol{c}_s - \boldsymbol{M}_{rs}\boldsymbol{b}_s - \boldsymbol{A}_{sr}^T\boldsymbol{M}_{ss}\boldsymbol{b}_s\right] + \left[\boldsymbol{T}_r + \boldsymbol{A}_{sr}^T\boldsymbol{T}_s\right]\boldsymbol{\tau} \end{aligned} \tag{3.17}$$

This can be written in a more succinct form:

$$\tilde{\boldsymbol{M}}\dot{\boldsymbol{u}}_r - \tilde{\boldsymbol{c}} = \tilde{\boldsymbol{T}}\boldsymbol{\tau} \tag{3.18}$$

3.4 Force Constraint

There is an alternate but equivalent formulation of the equations of motions which does not involve solving for the nonholonomic constraints, but rather imposes a force boundary condition at the tips to insure that the velocity at the tips of the arms is zero when the arms are grasping the bar. If Equation 3.7 is modified to include forces at the tip we get

$$\boldsymbol{M}\,\dot{\boldsymbol{u}} - \boldsymbol{c} = \boldsymbol{T}\boldsymbol{\tau} + \mathcal{T}\boldsymbol{f}_{tip} + \boldsymbol{f}_{external} \tag{3.19}$$

Where $\boldsymbol{f}_{tip}$ is a four vector representing the normal and tangential forces exerted by the bar on the arms at the tips, points $\boldsymbol{P}_5$ and $\boldsymbol{P}_8$.

$$\mathcal{T} = \begin{bmatrix} -\cos(q_3+q_4) & \sin(q_3+q_4) & -\cos(q_5+q_6) & \sin(q_5+q_6) \\ -\sin(q_3+q_4) & -\cos(q_3+q_4) & -\sin(q_5+q_6) & -\cos(q_5+q_6) \\ -L_{01_x}s_{34} + L_{01_y}c_{34} & -L_{01_x}c_{34} - L_{01_y}s_{34} & -L_{02_x}s_{56} + L_{02_y}c_{56} & -L_{02_x}c_{56} - L_{02_y}s_{56} \\ -L_{11}\sin(q_4) & -L_{11}\cos(q_4) & 0 & 0 \\ 0 & -L_{12} & 0 & 0 \\ 0 & 0 & -L_{21}\sin(q_6) & -L_{21}\cos(q_6) \\ 0 & 0 & 0 & -L_{22} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Rewriting Equation 3.14, yields

$$\boldsymbol{A}\boldsymbol{u} = 0 \tag{3.20}$$

$$\begin{bmatrix} 1 & 0 & -L_{01_y} & -L_{11}s_3 & -L_{12}s_{34} & 0 & 0 \\ 0 & 1 & L_{01_x} & L_{11}c_3 & L_{12}c_{34} & 0 & 0 \\ 1 & 0 & -L_{02_y} & 0 & 0 & -L_{21}s_5 & -L_{22}s_{56} \\ 0 & 1 & L_{02_x} & 0 & 0 & L_{21}c_5 & L_{22}c_{56} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = 0$$

Differentiating Equation 3.20 with respect to time in reference frame A yields

$$\boldsymbol{A}\dot{\boldsymbol{u}} = \boldsymbol{b} \tag{3.21}$$

$$b = \begin{bmatrix} u_1 u_2 + u_2{}^2 L_{01_x} + u_3{}^2 L_{11}\cos(q_3) + u_4{}^2 L_{12}\cos(q_3+q_4) \\ -u_0 u_2 + u_2{}^2 L_{01_y} + u_3{}^2 L_{11}\sin(q_3) + u_4{}^2 L_{12}\sin(q_3+q_4) \\ u_1 u_2 + u_2{}^2 L_{02_x} + u_5{}^2 L_{21}\cos(q_5) + u_6{}^2 L_{22}\cos(q_5+q_6) \\ -u_0 u_2 + u_2{}^2 L_{02_y} + u_5{}^2 L_{21}\sin(q_5) + u_6{}^2 L_{22}\sin(q_5+q_6) \end{bmatrix}$$

Using Equation 3.19 and Equation 3.21, the resultant forces exerted by the bar on the arms at points P_5 and P_8 are

$$f_{tip} = [AM^{-1}\mathcal{T}]^{-1}[b - AM^{-1}c - AM^{-1}T\tau] \tag{3.22}$$

3.5 Some Properties of the Center of Mass

In controlling the robot, there are some properties of the center of mass that are of interest. They are the position, velocity and acceleration of the center of mass. Also, the total angular momentum about the center of mass will be derived. The following definitions will be used in calculating the center of mass and momentum for the robot.

Definition 1 (Mass Center) *If S is a set of particles $P_1, \ldots, P_n$ of masses $m_1, \ldots, m_n$, respectively, there exists a unique point S^* such that*

$$\sum_{i=1}^{n} m_i \boldsymbol{r}_i = 0$$

where $\boldsymbol{r}_i$ is the position vector from S^* to P_i $(i = 1, \ldots, n)$. The point S^* is called the mass center.

Definition 2 (Linear Momentum) *If S is a set of particles $P_1, \ldots, P_n$ of masses $m_1, \ldots, m_n$, respectively, moving in a reference frame A with velocities ${}^A\boldsymbol{v}^{P_1}, \ldots, {}^A\boldsymbol{v}^{P_n}$, then*

$$\boldsymbol{L} = \sum_{i=1}^{n} m_i \, {}^A\boldsymbol{v}^{P_i}$$

where the vector $\boldsymbol{L}$ is called the total linear momentum.

Definition 3 (Angular Momentum) *If S is a set of particles $P_1, \ldots, P_n$ of masses $m_1, \ldots, m_n$, respectively, moving in a reference frame A with velocities ${}^A\boldsymbol{v}^{P_1}, \ldots, {}^A\boldsymbol{v}^{P_n}$, then the vector ${}^A\boldsymbol{H}^{S/S^*}$, called the angular momentum of S relative to S^* in A is defined as*

$${}^A\boldsymbol{H}^{S/S^*} = \sum_{i=1}^{n} m_i \boldsymbol{p}_i \times {}^A\boldsymbol{v}^{P_i}$$

where $\boldsymbol{p}_i$ is the position vector from the point S^* to P_i $(i = 1, \ldots, n)$.

The above definitions describe properties of a system of point masses, and do not deal with rigid bodies *per se*. However, for a system of rigid bodies, these definitions readily extend by treating each rigid body as behaving like a point mass at its mass center with a central inertia dyadic about the mass center. Then definitions can be stated in a more useful form as:

$$\boldsymbol{L} = \sum_{i=1}^{\text{All bodies}} m_i \, {}^A\boldsymbol{v}_i^* \tag{3.23}$$

$${}^A\boldsymbol{H}^{S/S^*} = \sum_{i=1}^{\text{All bodies}} \mathbf{I}_i^* \cdot \boldsymbol{\omega}_i + m_i \boldsymbol{p}_i \times {}^A\boldsymbol{v}_i^* \tag{3.24}$$

where all points are taken to the mass centers of each of the bodies.

With these definition, the mass center and momentum of the robot can be calculated as a function of the state, masses, and lengths.

$$\begin{aligned} \boldsymbol{x}^{\text{CM}} &= q_0\boldsymbol{a}_1 + q_1\boldsymbol{a}_2 \\ &\quad + (z_{11} - z_3c_3 + z_4s_3 - z_5c_{34} + z_6s_{34} - z_7c_5 + z_8s_5 - z_9c_{56} + z_{10}s_{56})/z_1\boldsymbol{b}_1 \\ &\quad - (z_2 + z_3s_3 + z_4c_3 + z_5s_{34} + z_6c_{34} + z_7s_5 + z_8c_5 + z_9s_{56} + z_{10}c_{56})/z_1\boldsymbol{b}_2 \\ &= q_0\boldsymbol{a}_1 + q_1\boldsymbol{a}_2 + \frac{1}{z_1}\left[\left\{\sum_{i=2}^{7} M(1,i)\right\}\boldsymbol{b}_1 - \left\{\sum_{i=2}^{7} M(0,i)\right\}\boldsymbol{b}_2\right] \end{aligned} \tag{3.25}$$

The velocity of the center of mass is given by:

$$\begin{aligned} {}^A\boldsymbol{v}^{\text{CM}} &= (z_1u_0 + z_2u_2 + (z_3s_3 + z_4c_3)u_3 + \\ &\quad (z_5s_{34} + z_6c_{34})u_4 + (z_7s_5 + z_8c_5)u_5 + (z_9s_{56} + z_{10}c_{56})u_6)/z_1\boldsymbol{b}_1 \\ &\quad + (z_1u_1 + z_{11}u_2 - (z_3c_3 - z_4s_3)u_3 - \\ &\quad (z_5c_{34} - z_6s_{34})u_4 - (z_7c_5 - z_8s_5)u_5 - (z_9c_{56} - z_{10}s_{56})u_6)/z_1\boldsymbol{b}_2 \\ &= \frac{1}{z_1}\left[\left\{\sum_{i=0}^{7} M(0,i)\boldsymbol{u}(i)\right\}\boldsymbol{b}_1 + \left\{\sum_{i=0}^{7} M(1,i)\boldsymbol{u}(i)\right\}\boldsymbol{b}_2\right] \end{aligned} \tag{3.26}$$

The total linear momentum of the system ${}^A\boldsymbol{L}^{\text{CM}}$ is $z_1\boldsymbol{v}^{\text{CM}}$.

The acceleration of the center of mass is given by:

$${}^A\boldsymbol{a}^{\text{CM}} = \frac{1}{z_1}\left[\left\{\sum_{i=0}^{7} M(0,i)\dot{\boldsymbol{u}}(i) - \boldsymbol{c}(0)\right\}\boldsymbol{b}_1 + \left\{\sum_{i=0}^{7} M(1,i)\dot{\boldsymbol{u}}(i) - \boldsymbol{c}(1)\right\}\boldsymbol{b}_2\right] \tag{3.27}$$

The central angular momentum, taken about the mass center, is given by:

$$ {}^{A}\boldsymbol{H}^{\mathrm{CM}} = \sum_{i=2}^{7}\sum_{j=0}^{7} M(i,j)\boldsymbol{u}(j)\,\boldsymbol{b}_3 + z_1\boldsymbol{p}^{\mathrm{CM}} \times \boldsymbol{v}^{\mathrm{CM}} \tag{3.28} $$

where ${}^{A}\boldsymbol{p}^{\mathrm{CM}}$ is a vector from the center of mass of the base body B to the system center of mass and is given as ${}^{A}\boldsymbol{p}^{\mathrm{CM}} = \boldsymbol{x}^{\mathrm{CM}} - q_0\boldsymbol{a}_1 - q_1\boldsymbol{a}_2$.

Finally, the relationship between the derivative of momentum and the generalized accelerations can be shown to be:

$$ \begin{bmatrix} \dot{\boldsymbol{L}} \cdot \boldsymbol{b}_1 \\ \dot{\boldsymbol{L}} \cdot \boldsymbol{b}_2 \\ \dot{\boldsymbol{H}}_z^{\mathrm{CM}} \cdot \boldsymbol{b}_3 \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^{7} M_{0j}\dot{u}_j - c_0 \\ \sum_{j=0}^{7} M_{1j}\dot{u}_j - c_1 \\ \sum_{i=2}^{7}\sum_{j=0}^{7} M_{ij}\dot{u}_j - c_i \end{bmatrix} \tag{3.29} $$

Once the equations of motions are derived, the kinetic energy is easily calculated as

$$ \text{Kinetic Energy} = \frac{1}{2}\boldsymbol{u}^T M \boldsymbol{u} \tag{3.30} $$

Note that all the quantities of interest are given in terms of the mass matrix. From Equation 3.8, it is apparent that the mass matrix can be constructed with only 14 additions and 28 multiplications.

Chapter 4

Strategic Control

This chapter describes the high level strategic controller. The strategic controller provides a means of performing task level commands. It provides an interface to low-level control laws which control physical states of the system. This chapter introduces the concept of state diagrams for designing and programming the strategic controller. An advantage of a strategic controller is that it allows for the rapid switching of control laws during the execution of a task. This is a very desirable capability for a system undergoing abrupt kinematic constraints.

4.1 Introduction

The classic digital control paradigm is depicted in Figure 4.1. This controller executes synchronously and serially, making it easily implementable on digital computers. These dynamic controllers are the heart of any robot control scheme, and are discussed in detail in Chapter 5. Unfortunately, dynamic controllers do not provide an easy interface to task level commands. Being error based, most dynamic controllers accept as inputs a *desired state* and not a *desired task*. Therefore, a higher level of abstraction is needed to interface desired tasks to desired states. The strategic controller is one such interface. Many of the ideas for strategic control come from the successful work of Schneider [27, 28] in applying strategic control to cooperative manipulation.

By way of example, consider the task of "leaping across the table" as shown in Figure 4.2. If an initial angle and velocity is given at the time of release, then the task is well defined. The reason this task is awkward to implement in a single dynamic controller is because discrete changes in the plant are occuring asynchronously. For example, when

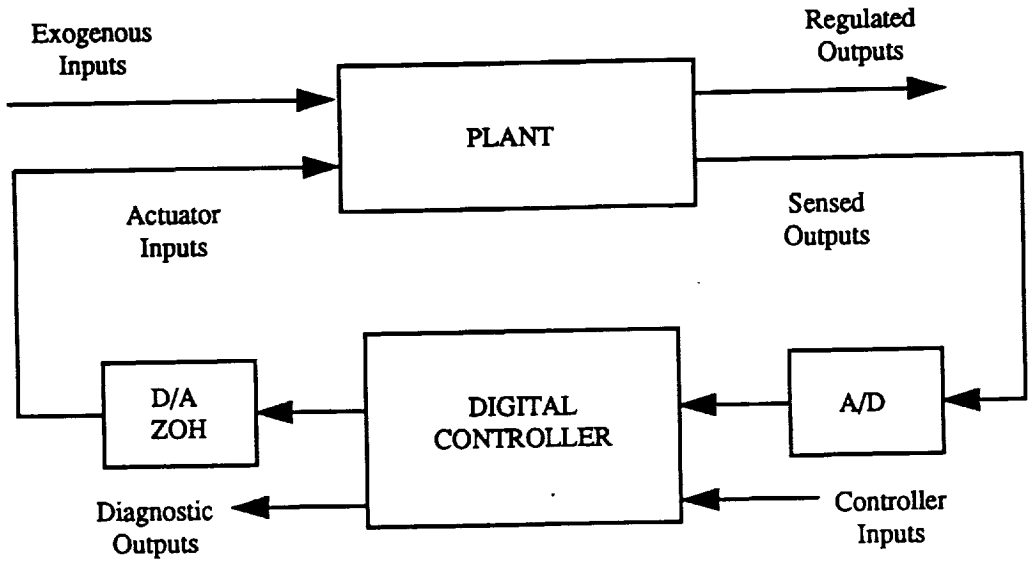

Figure 4.1: **Classic Digital Controller**

A block diagram of the classic digital controller. The inputs may not be in a convenient form for task level control.

both arms release the bar, the number of degrees of freedom of the system are increased by four. Furthermore, tasks might require a PD controller, bang-bang controller, a variation of a computed torque controller, or some combination of the above.

4.2 State Transition Graphs

State transition graphs or *state diagrams* [10] are one approach to designing a strategic controller. In this context, one graphically divides a desired task into a set of phases or finite states. To complete a task or operation, states must be transitioned in a predefined order. The states need not be transitioned sequentially or only once, but can involve loops as shown in Figure 4.4. The transition between states can only occur in response to an external event or stimulus. The event and the current state uniquely determines the next state.

For example, consider the state diagram shown in Figure 4.3. Each state, represented pictorially as an oval, must be transitioned in order for the robot to complete a LEAP maneuver. The arrows correspond to external events. The robot starts off in the "Idle"

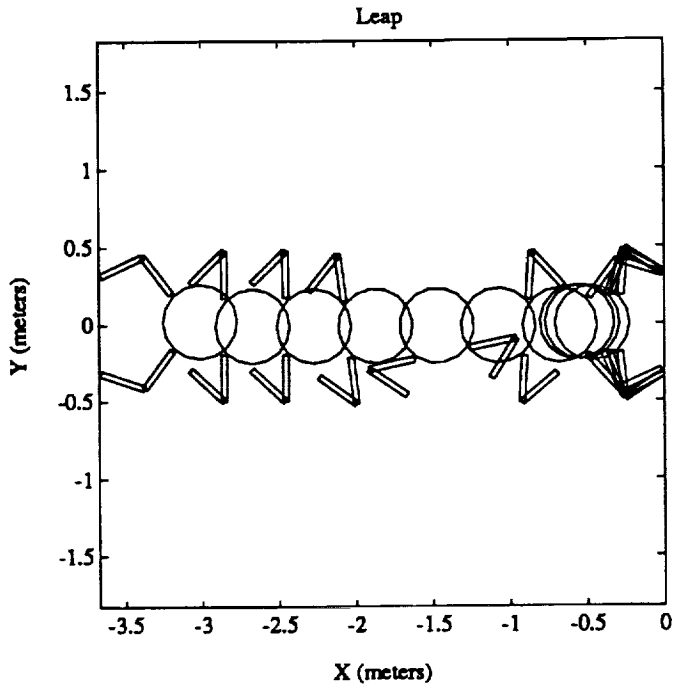

Figure 4.2: **Coordinated Leap Maneuver**

The figure shows the Pushoff, Rotation, and Catch phases for a straight back leap. The robot is going from right to left.

state. Receiving a request to leap causes the robot to enter the "Windup" state. A "Ready" stimulus is sent (from a different process) when joint angles and joint rates are within predetermined tolerances, and the "Pushoff" state is entered. The system imparts momentum to itself by pushing off the bar along a predefined momentum trajectory.

At this point two different events are possible. One possibility is that the system has achieved the desired final momentum, and so the "Coast" state is entered. The other possibility is that an error occured, such as a timeout warning or a gripper failing to open. In this case the "Idle" state is entered and the robot attempts error recovery. If all goes well, the robot "Tucks" in its arms to reduce its moment of inertia and rotates via the momentum wheel. Finally the system transitions between the "Approach" state and the "Catch" state when both grippers grasp the bar. After grabbing the bar, the system is brought to rest (zero momentum state) and remains in the "Idle" state.

Another example is a crawl task, as shown in Figure 4.4. As one can see from these two simple examples, complicated locomotion tasks can be easily visualized by drawing state

transition graphs.

Also, error conditions are easily incorporated into the task specification. More complicated error conditions and tasks can also be designed with this technique. The tasks

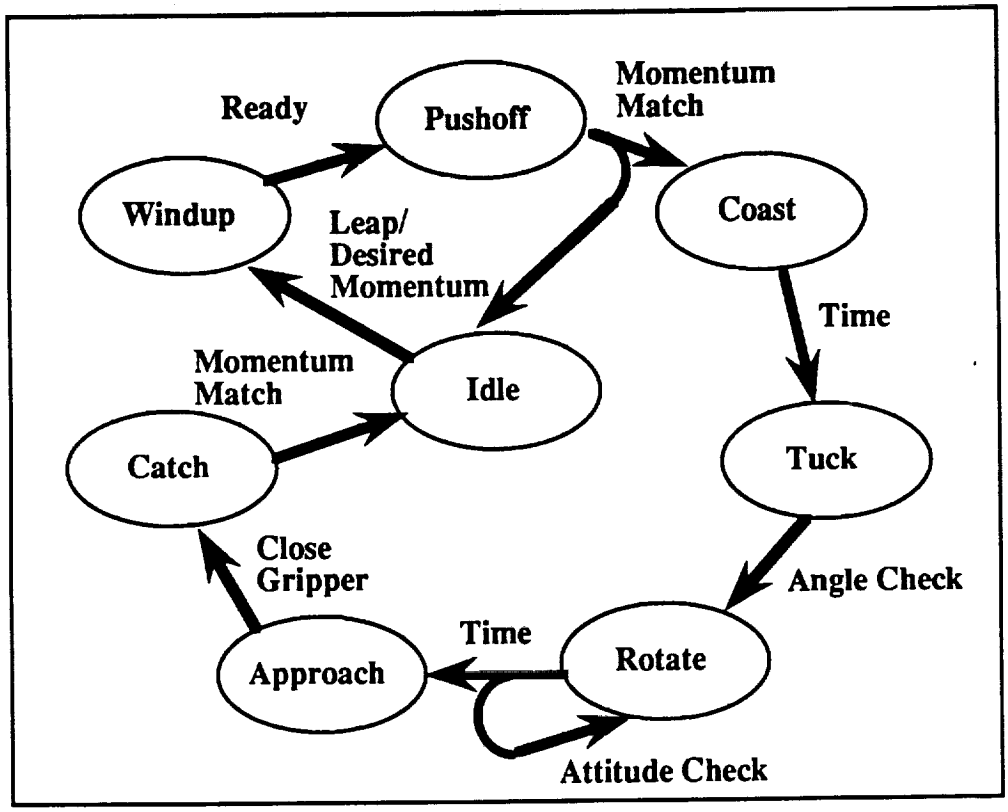

Figure 4.3: **State Transition Diagram for a Leap Maneuver**

described in the state diagrams are implemented in a strategic controller by using state table programming. Table 4.1 is the corresponding state transition table for the state diagram shown in Figure 4.3. Each input or stimuli determines how the system will transition from one state to another. After receiving a stimulus, a transition routine[1] is executed which returns the value for the next state. For example, if the robot is in the "Idle" state and receives a "Leap" stimulus, the routine "CheckConfig" will be executed. If the robot is not grabbing the bar, it cannot perform a pushoff, and an error condition is detected. The routine returns the robot to the idle state with a warning message. If no error condition is detected, a windup trajectory is planned and executed. Upon successfully exiting the "CheckConfig" routine, the robot enters the "Windup" state.

[1] In the actual implementation, each state is assigned a value. A transition routine is a C function that returns the value for the next state.

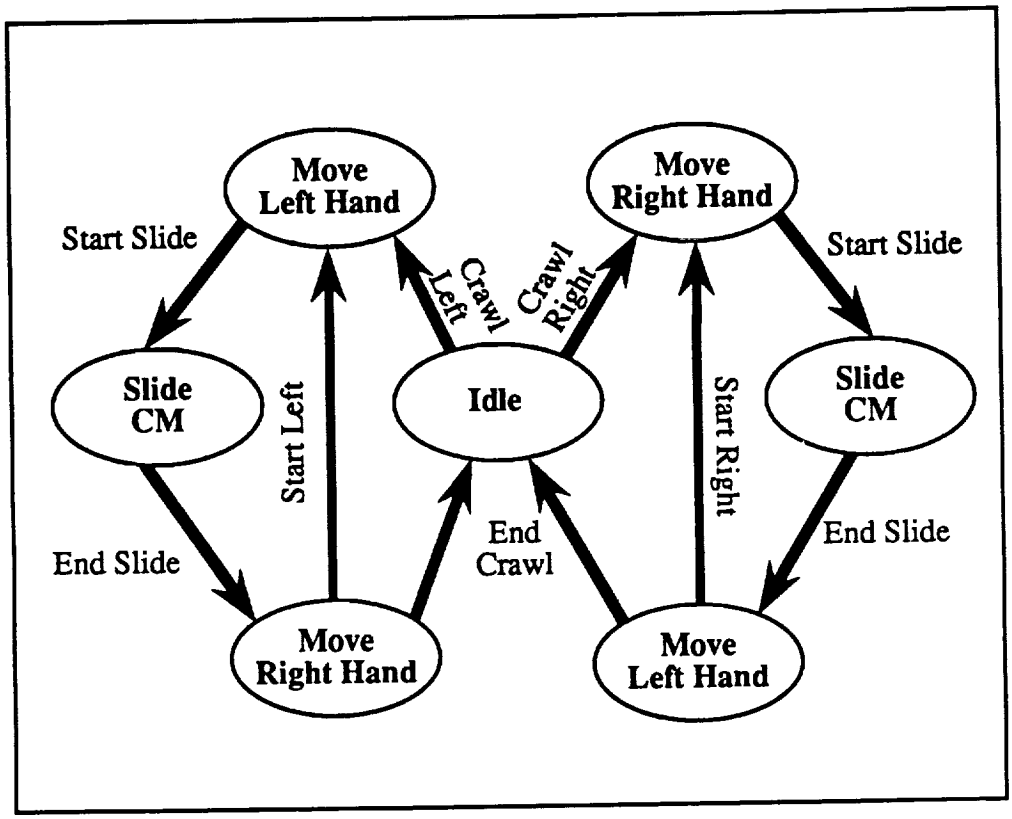

Figure 4.4: **State Transition Diagram for a Crawl Maneuver**

Stimuli can come from a variety of external sources, but never from the strategic controller itself. They are, in fact, the input to the controller. In general, they originate from one of three places:

1. *command* stimuli from the user, such as a Leap stimulus.
2. *helper* stimuli, which come from external processes that monitor the system and report when a condition or event occurs. For example, a stimulus is sent to the strategic controller when the momentum reaches a desired value.
3. *hardware* stimuli, such as watchdog timers.

In the event that an erroneous or unexpected stimulus occurs, a default error handling routine is executed.

State	Stimulus	Transition Routine	Next States	
Idle	Leap	CheckConfig	WindUp	Idle
Windup	Ready	CalcLeapTrajectory	PushOff	
PushOff	MomentumMatch	Release	Cost	
	Time	SafeState	Idle	
	Range	SafeState	Idle	
Coast	Release	ArmTuck	Tuck	Idle
Tuck	Time	PID Control	Tuck	
	Ready	ActivateWheel	Rotating	Idle
Rotating	StopRotating	SetApproach	Approach	
	Time	SetApproach	Approach	
Approach	Grab	GraspBar	Catch	
Catch	Time	Dock	Idle	

Table 4.1: **State Transition Table**

4.2.1 Implementation

The following code fragments show the data structure and declaration of the state table. All code is written in ANSI C using the GNU C compiler.

```
#define NAMELEN  (32)
#define MAXSTIMULI (16) /* Max stimuli branches in one state. */
#define MAXNEXTSTATES    (5) /* Max next states for one stimulus. */


/* Data structure definition for a "state" */
typedef struct TaskStateTag {
    char name[NAMELEN];
    struct StimulusResponseTag {
        int stimulus;           /* This entry's stimulus */
        int (*proc)();          /* The associated transition routine. */
        struct TaskStateTag *nextState[MAXNEXTSTATES];  /* The set of next
                                                         * states */
    }   sr[MAXSTIMULI];
} TaskStateStorage, *TaskState;


typedef struct StimulusResponseTag
    *StimulusResponse, StimulusResponseStorage;
```

```
/* Forward declaration of the states */
static TaskStateStorage Idle,WindUp,PushOff,Coast,
                        Tuck,Rotating,Approach,Catch;
```

Having defined the data structure, it is a simple matter to transfer the information from the state transition table into C. For the Leap task, the code looks like the following:

```
/* Leap chain */
static TaskStateStorage Idle = {"Idle",
    {
        {LEAP_STIM, CheckConfig, {&WindUp, &Idle, &ReGrasp}},
        {TIME_STIM, SafeState, {&Idle}},
        {DEMO_STIM, Thrusting, {&Coast}},
        {OUTOFRANGE_STIM, SafeState, {&Idle}},
        {0}
    }
};

static TaskStateStorage WindUp = {"WindUp",
    {
        {TIME_STIM, CalcLeapTrajectory, {&PushOff, &Idle}},
        {READY_STIM, CalcLeapTrajectory, {&PushOff, &Idle}},
        {OUTOFRANGE_STIM, SafeState, {&Idle}},
        {0}
    }
};
```

4.3 Trajectory Generation

Once the strategic controller is programmed to perform a task, trajectories must be generated for the dynamic controller. In order to implement a computed torque or inverse dynamic control law, the trajectories must be continuous in acceleration. Of course the term "trajectory" is used here in a very broad sense to describe a generic path. For example, one could generate joint trajectories, Cartesian endpoint trajectories, or momentum trajectories.

Currently, all the trajectories in this research are generated by fifth-order polynomial

splines. There are theoretical [22, 37] as well as experimental [13, 8] studies which show that fifth-order polynomial give a good combination of slew time, smooth changes in acceleration, and minimum computation. One motivation for selecting fifth-order splines is that they are the set of functions that minimize the mean-square jerk[2] over the trajectory. Work by Flash and Hogan [8] suggests that humans instinctively follow minimum jerk trajectories while performing pick and place tasks. The following is a derivation to show that fifth-order polynomials do indeed minimize the mean-squared jerk.

Let F be a function that has continuous second partial derivatives with respect to its arguments. We wish to find the function $x(t)$ that minimizes the expression:

$$I = \int_{t_0}^{t_1} F(t, x, x', x'', x''')dt \tag{4.1}$$

where x' denotes differentiation of x with respect to t. To minimize I, a necessary condition is for the first variation to vanish [11]. This can be written as:

$$\delta I = \int_{t_0}^{t_1} [\frac{\partial F}{\partial x}\delta x + \frac{\partial F}{\partial x'}\delta x' + \frac{\partial F}{\partial x''}\delta x'' + \frac{\partial F}{\partial x'''}\delta x''']dt = 0$$

Integrating the last term by parts yields:

$$\delta I = \int_{t_0}^{t_1} [\frac{\partial F}{\partial x}\delta x + \frac{\partial F}{\partial x'}\delta x' + \frac{\partial F}{\partial x''}\delta x'' - \frac{d}{dt}(\frac{\partial F}{\partial x'''})\delta x'']dt + \left[\frac{\partial F}{\partial x'''}\delta x''\right]_{t_0}^{t_1} = 0$$

Integrating twice more by parts gives:

$$\begin{aligned} \delta I &= \int_{t_0}^{t_1} [\frac{\partial F}{\partial x} - \frac{d}{dt}(\frac{\partial F}{\partial x'}) + \frac{d^2}{dt^2}(\frac{\partial F}{\partial x''}) - \frac{d^3}{dt^3}(\frac{\partial F}{\partial x'''})]\delta x dt \\ &+ \left[\frac{\partial F}{\partial x'''}\delta x''\right]_{t_0}^{t_1} + \left[(\frac{d}{dt}\frac{\partial F}{\partial x'''} - \frac{\partial F}{\partial x''})\delta x'\right]_{t_0}^{t_1} + \left[(\frac{d^2}{dt^2}\frac{\partial F}{\partial x'''} - \frac{d}{dt}\frac{\partial F}{\partial x''} + \frac{\partial F}{\partial x'})\delta x\right]_{t_0}^{t_1} = 0 \end{aligned} \tag{4.2}$$

Any function $x(t)$ that minimizes Equation 4.1 must satisfy the *Euler equation*

$$\frac{\partial F}{\partial x} - \frac{d}{dt}(\frac{\partial F}{\partial x'}) + \frac{d^2}{dt^2}(\frac{\partial F}{\partial x''}) - \frac{d^3}{dt^3}(\frac{\partial F}{\partial x'''}) = 0 \tag{4.3}$$

With three natural boundary conditions:

$$\left[\frac{\partial F}{\partial x'''}\delta x''\right]_{t_0}^{t_1} = 0 \tag{4.4}$$

$$\left[(\frac{d}{dt}\frac{\partial F}{\partial x'''} - \frac{\partial F}{\partial x''})\delta x'\right]_{t_0}^{t_1} = 0 \tag{4.5}$$

$$\left[(\frac{d^2}{dt^2}\frac{\partial F}{\partial x'''} - \frac{d}{dt}\frac{\partial F}{\partial x''} + \frac{\partial F}{\partial x'})\delta x\right]_{t_0}^{t_1} = 0 \tag{4.6}$$

[2] Jerk is the derivative of acceleration.

Note that if the unknown function $x(t)$ and its derivatives have prescribed values at the endpoints, the three natural boundary conditions are satisfied.

We are now ready to solve the for the minimum jerk. This can now be stated as:

$$\text{minimize } \mathcal{J} = \int_0^{t_f} \left(\frac{d^3x}{dt^3}\right)^2 dt$$

given $x(0), x'(0), x''(0), x(t_f), x'(t_f)$, and $x''(t_f)$.

Since the conditions at the endpoints are given, the three natural boundary conditions are satisfied. Therefore, the Euler equation becomes:

$$\frac{d^3}{dt^3}\left(\frac{\partial}{\partial x'''}\left[\frac{d^3x}{dt^3}\right]^2\right) = 0$$

or

$$\frac{d^6x}{dt^6} = 0 \tag{4.7}$$

$$x(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \tag{4.8}$$

which is the desired fifth-order polynomial. The coefficients a_i are uniquely determined by the boundary conditions and are given by:

$$\begin{aligned}
a_0 &= x(0) \\
a_1 &= x'(0) \\
a_2 &= \frac{x''(0)}{2} \\
a_3 &= \frac{(x''(t_f) - 3x''(0))t_f^2 - (8x'(t_f) + 12x'(0))t_f + 20(x(t_f) - x(0))}{2t_f^3} \\
a_4 &= \frac{-(2x''(t_f) - 3x''(0))t_f^2 + (14x'(t_f) + 16x'(0))t_f - 30(x(t_f) - x(0))}{2t_f^4} \\
a_5 &= \frac{(x''(t_f) - x''(0))t_f^2 - 6(x'(t_f) + x'(0))t_f + 12(x(t_f) - x(0))}{2t_f^5}
\end{aligned}$$

Momentum Trajectory Generation Momentum trajectories are used to specify the position and velocity of the mass center of the robot. For example, while the robot is at rest in the closed-chain configuration, it could be commanded to "move its mass center 0.15 m in the right". Since forces are linearly related to the first derivative of momentum, a fifth-order polynomial is sufficient to specify the path of the mass center through momentum space. The coefficients of the polynomial are determined by specifying the elapsed time of

the slew, and the initial and final position of the mass center. Since the mass center is at rest at the beginning and end of the slew, all higher order derivative terms are set to zero.

In general, one calculates a momentum trajectory by specifying a time interval, and initial and final momentum states. A momentum state consists of five vectors: the integral of linear momentum, both linear and angular momentum, and the derivatives of linear and angular momentum. In two dimensional Cartesian space, this corresponds to eight distinct quantities to specify uniquely a momentum state. These desired momentum states cannot be chosen arbitrarily, for there are constraints on the system such as the location of the mass center when both arms are grabbing the bar.

The vehicle performs a windup maneuver so as to accelerate the vehicle along the longest available path. The final momentum state at the end of the "windup" becomes the initial momentum state during "pushoff". Empirically, it was found that the center of mass can move along a ray for 0.2 m (Δ x = .2) with a final velocity of 0.1 m/s. The trajectories were formulated such that the acceleration along the ray was constant. This assured that all the energy of the vehicle was directed toward *imparting* momentum to the system during "pushoff". With these assumptions and a desired release angle[3] θ, the final momentum state can be uniquely determined. Figure 4.5 is a pictorial representation of a momentum trajectory.

4.4 The Switching Problem

When the strategic controller receives a stimulus that an event occurred, such as a change in the number of degrees of freedom of the system, it should react to this discontinuous change in the plant in such a way as to achieve smooth transitions. The following description explains why it should always be possible to achieve this transition with rigid-body manipulators.

While in the closed-chain configuration, the number of degrees of freedom is reduced because of kinematic constraints. Differentiating the constraint equations gives a linear constraint in the velocities, or generalized speeds, as was shown in Equation 3.14. Thus, it is always possible to compute a consistent set of generalized coordinates and generalized speeds for a rigid body system. When transitions occur, they must be continuous in position, and can only be discontinuous in velocity as a result of impulsive forces. This discontinuity in velocity can occur when the tips of the arms grab the bar with a non-zero

[3]The angle $\theta = 0$ rad corresponds to a straight-back pushoff as shown in Figure 6.1.

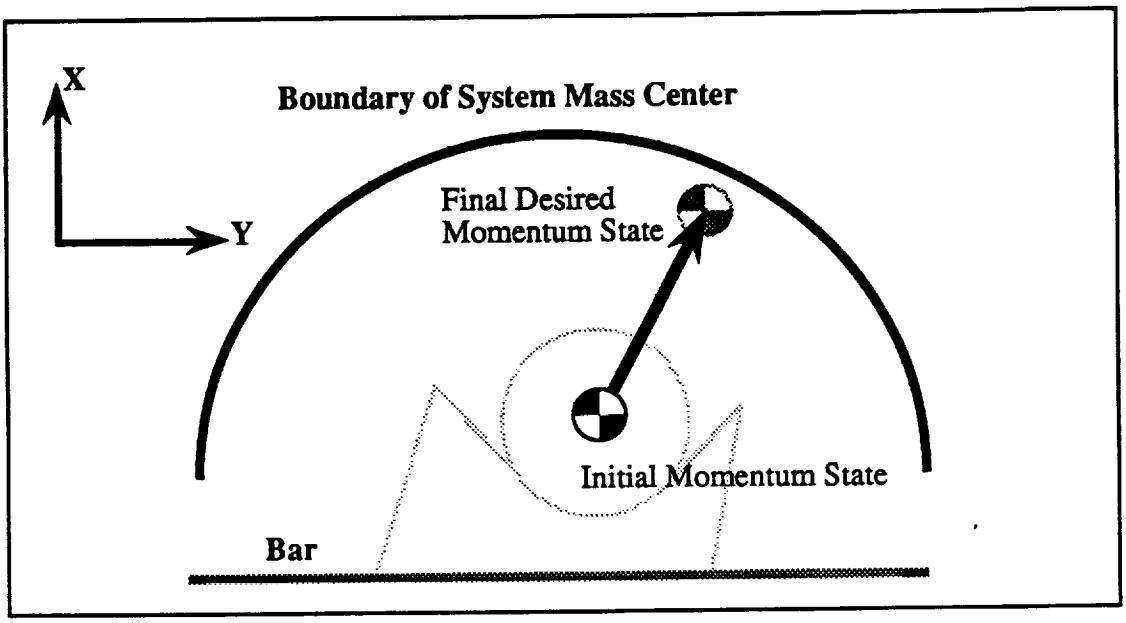

Figure 4.5: **A Momentum Trajectory**

The arrow represents a possible path of the mass center. To move the mass center along a path, a fifth-order trajectory is generated from Equation 4.8. The coefficients are determined by specifying the position of the mass center, momentum, and derivative of the momentum at the two endpoints of the trajectory. The position of the mass center is constrained by the lengths of the arms, while the velocity is constrained by link lengths and maximum torque capability.

velocity and decelerate to zero in "zero time". Of course the state does not go to zero: only the velocity at each tip does.

In the case of a "soft landing", the relative velocity of the end effector and the target is zero, and so all the states are continuous during the transition. By calculating a consistent set of states for the full holonomic system, any controller will have the current estimate of the state. Thus, when switching from one control law to another, errors in configuration will not contribute to transient errors in the behavior of the system.

4.5 The Strategic Controller

The strategic controller is executed as a separate process by the real-time operating system. Most of the time it is blocked on a read, meaning that the processes is idle and waiting for

input. All inputs to the finite-state machine are in the form of stimuli. After a stimulus is received, a corresponding transition or subroutine is executed. These routines check on the physical configuration (open-chain or closed-chain) of the robot, and implement a new model of the system when necessary. The transition routines can also install new filters and controllers, as well as change gains in the dynamic controller.

Although there are many advantages in this implementation, it is not without drawbacks and limitations. Because the finite-state machine runs as its own process, the controller has more than one thread of execution. This presents problems for real-time debugging and offline simulation. Also, the stimuli arrive asynchronously, yet must be incorporated with synchronous control. Modern operating systems, such as VxWorks, provide a message-passing facility that performs the synchronization.

4.5.1 Implementation

The following code segment is an ANSI C version of the finite-state machine. When a stimulus is received, the finite-state machine compares the stimulus with a list of allowed ones for the current state. It then executes the transition routine, and moves to the next state based on the transition routine's return value.

```
static TaskState State;
static int FSMxid;
boolean FsmVerbose;

static void FSMDriver(int stimulus)
{
    register boolean found = FALSE;
    register StimulusResponse sr;
    register int stimno, retcode;
    register TaskState lastState;

    lastState = State;

    stimno = 0;

    /* Look up the stimulus */
    for(stimno = 0; ; stimno++) {
```

```
        sr = &State->sr[stimno];
        if(sr->stimulus == 0) {
            if(FsmVerbose) {
                printf("FSMDriver: Unexpected stimulus %d.\n",stimulus);
            }
            found = FALSE;
            break;
        }
        if((sr->stimulus == stimulus) || (sr->stimulus == ANY_STIM)) {
            found = TRUE;
            break;
        }
    }

    if(found) {
        retcode = (*sr->proc)(stimulus); /* Execute the transition routine. */
        State = sr->nextState[retcode];  /* Update the State */
    }
    if(FsmVerbose) {
        printf("FSM: %s --%2d--> %s\n", lastState->name, stimulus, State->name);
    }
}

/* This daemon exists simply to weave the asynchronous stimuli into a
   synchronous stream.  Note also that it allows the stimulus routines
   to run Unlocked, thus preventing unnecessarily long interrupt
   latency. */
static void FSMDaemon ()
{
    int stimulus;

    while(TRUE) {
        read(FSMxid, &stimulus, sizeof(int));   /* Wait for a stimulus. */
        FSMDriver(stimulus);                    /* Process it. */
```

```
    }
}

/* Send a stimulus to the state machine.  WARNING!  This code MUST be
   re-entrant, as it is executed asynchronously by routines of varying
   priority.  */
static void SendFSMStimulus(int stimulus)
{
    write(FSMxid, &stimulus, sizeof(int));
}
```

Chapter 5

Dynamic Control

This chapter describes the various dynamic controllers used in relocating and reorienting the robot. Because the system undergoes changes in configuration, different controllers are implemented at different times depending upon the state of the strategic controller described in Chapter 4. For locomotion, there are two basic quantities of interest to control: momentum and orientation. Other tasks, such as manipulation, control endpoint position or endpoint force. By formulating the equations of motion as shown in Chapter 3, the strategic controller can easily switch amongst a variety of control laws which regulate different quantities. The basic controllers discussed in this chapter are:

1. PD and PID controllers to regulate joint angles. This is a robust but crude control law used to tuck in the arms while the robot is controlling its orientation.

2. Momentum controller to cause the system to follow desired *momentum trajectories* discussed in Chapter 4. This controller is used to impart momentum to the system during a pushoff maneuver or to relocate the mass center during a crawl maneuver.

3. Bang-Bang controller to reorient the attitude of the robot in minimum time.

This chapter outlines a technique for controlling the momentum of a robotic system. This is accomplished by causing the linear and angular momentum to follow desired momentum trajectories in the presence of kinematic constraints, which occur in closed-kinematic-chain configurations. This powerful idea facilitates a simple interface between the high-level strategic controller and the low-level dynamic controller that is "exact" in that it compensates for all dynamic forces. This control paradigm does not involve inversion of the mass matrix and is amenable to parallel processing, even though the resulting system is not modeled as an "in-parallel system". The system momentum controller has been implemented

on a free floating robot to perform such maneuvers as pushoff and catch.

In addition, this chapter discusses some of the issues involved in orientation of a free-flying multi-link system of rigid bodies. While the robot is free-flying, the momentum is uncontrollable using internal torques and forces as a consequence of the principle of conservation of momentum. However, conservation of angular momentum results in non-holonomic constraint equations. It is through these equations that the *orientation* of the robot can be controlled.

5.1 PD and PID controllers

Proportional-derivative (PD) and proportional-integral-derivative (PID) control is the classic standard control law by which new control laws are judged. Any text on feedback control such as Franklin and Powell [9] will describe various design techniques for choosing the velocity and position gains. Its major advantages are simplicity in implementation and guaranteed stability for passive systems (linear and nonlinear) in the presence of parameter variations. The disadvantage is that no information on the model is incorporated into the controller, and so performance is usually not as good as with other methods. More importantly, it is difficult to incorporate abrupt kinematic changes into a PD design. To accomplish this, a new controller, called system momentum control was developed.

5.2 Computed-Torque Control

Computed-torque control or inverse-dynamics control is one type of nonlinear controller design that has gained popularity for control of rigid robot manipulators. In implementing a computed torque controller, the control law is decomposed into two parts: a model-based control law that linearizes and decouples the plant equations, and an error-driven or servo control law that is formed by differencing desired and actual values [6]. A more general approach to linearizing control of nonlinear systems can be be found by Sastry and Bodson [26]. The following sections describe three different variations of computed torque: joint-based computed torque, Cartesian-based computed torque, and system momentum control.

Joint-based computed torque For any n degree of freedom rigid-body manipulator, the dynamics have the form:

$$T\tau = M(q)\dot{u} - c(q,u) - f(q,u)_{external} \tag{5.1}$$

$$u = Y(q)\dot{q}$$

where the inertia matrix is $M \in R^{n \times n}$, $c \in R^n$ is a vector of centrifugal and Coriolis terms, and $f_{external} \in R^n$ is a vector of other forces such as gravity, friction, and external disturbances. The problem of controlling a nonlinear system such as Equation 5.1 can be accomplished if one partitions the control law into two parts:

$$\tau = \alpha\tau' + \beta$$

where $\tau \in R^n$ is a vector of joint torques. By choosing

$$\alpha = T^{-1}M \tag{5.2}$$

$$\beta = -T^{-1}\{c + f_{external}\} \tag{5.3}$$

The servo law becomes

$$\ddot{q}_c = \ddot{q}_d + K_v(\dot{q}_d - \dot{q}) + K_p(q_d - q) \tag{5.4}$$

$$\tau' = Y\ddot{q}_c + \dot{Y}\dot{q} \tag{5.5}$$

where $\ddot{q}_c$ is the commanded acceleration to the control law and $\ddot{q}_d$ is the desired acceleration. In the absence of any trajectory error, these two quantities would be equal. Figure 5.1 shows a block diagram of the computed-torque control scheme. If the state is chosen to be q and $\dot{q}$, with the forces acting at the joints, and $T = Y = I$, then Equations 5.2-5.5 become the familiar form found in Craig [6].

Example: Joint-based computed torque for two arm floating robot This example shows how to design a computed torque controller for a two arm robot on a floating base. In this case, there are 8 position and velocity states, but only 5 motors. By partitioning the state into controllable and uncontrollable parts, one can resolve the torques necessary to control the quantities of interest as follows. Let:

$$q_u \triangleq \begin{bmatrix} q_0 \\ q_1 \\ q_7 \end{bmatrix} \quad q_c \triangleq \begin{bmatrix} q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} \quad u_u \triangleq \begin{bmatrix} u_0 \\ u_1 \\ u_7 \end{bmatrix} \quad u_c \triangleq \begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix}$$

$$\begin{bmatrix} M_{cc} & M_{cu} \\ M_{cu}^T & M_{uu} \end{bmatrix} \begin{bmatrix} \dot{u}_c \\ \dot{u}_u \end{bmatrix} = \begin{bmatrix} c_c \\ c_u \end{bmatrix} + \begin{bmatrix} T_c \\ T_u \end{bmatrix} \tau \tag{5.6}$$

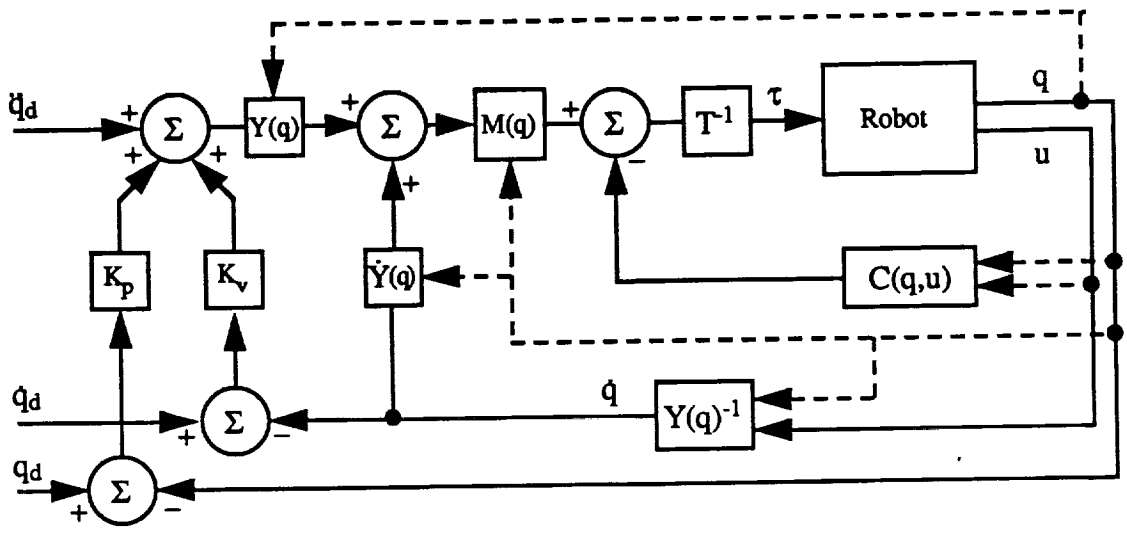

Figure 5.1: **Computed Torque**

Block diagram of a computed torque controller for a general rigid body robot. Dashed lines represent a nonlinear relation between input and output.

The above equation is just a reordering of the rows of the equations of motion. By rearranging terms, the vector $\dot{u}_u$ can be solved for in terms of measured or desired quantities.

$$\dot{u}_u = M_{uu}^{-1}[c_u + T_u\tau - M_{cu}^T\dot{u}_c] \tag{5.7}$$

By recognizing that the term $M_{cu}M_{uu}^{-1}T_u$ is zero for the free-flying LEAP system, the computed torque equation becomes:

$$\tau = T_c^{-1}\{[M_{cc} - M_{cu}M_{uu}^{-1}M_{cu}^T]\dot{u}_c + [M_{cu}M_{uu}^{-1}c_u - c_c]\} \tag{5.8}$$

Everything is known to calculate τ except $\dot{u}_c$. To calculate $\dot{u}_c$ let:

$$\dot{u}_c = Y_c\{\ \ddot{q}_d + K_v(\dot{q}_d - \dot{q}_c) + K_p(q_d - q_c)\} \tag{5.9}$$

The major difference between a fixed-base and free-flying robot is the singularity of the T matrix. In the case of free-flying robots, the momentum of the system cannot be controlled by internal motors, and so the number of actuators is less than the number of states. If one uses thrusters to control momentum, the two cases can be treated the same.

Cartesian-based Control In many situations, one is interested in specifying a trajectory of the tip of a manipulator, and not the motion of the joints themselves. In this case it is more convenient to formulate the control law in *Cartesian space* instead of *Joint Space.* Such a control is also called Operational-Space Control [1, 18].

Starting from the basic equations of motion given in Equation 5.1

$$\begin{aligned} T\tau &= M(q)\dot{u} - c(q,u) - f(q,u)_{external} \\ v &= J(q)u \end{aligned} \tag{5.10}$$

where $v \in R^n$ is a vector of translational and rotational velocities of the end-effector in Cartesian space and $J(q) \in R^{n\times n}$, called the Jacobian, is a matrix that relates generalized velocities u in joint-space to Cartesian-space velocities v at the end-effector. By differentiating Equation 5.10 and substituting into Equation 5.1,

$$\begin{aligned} a &= \dot{v} = \dot{J}(q,u)u + J(q)\dot{u} \\ &= \dot{J}(q,u)u + J(q)M^{-1}(q)\left\{c(q,u) + f(q,u)_{external} + T\tau\right\} \end{aligned} \tag{5.11}$$

where a is a vector of accelerations at the tip. Note that the derivative $\dot{J}(q,u)$ of the Jacobian is a function of both q and u because of the linear dependence between $\dot{q}$ and u. The commanded torques at the joints can be derived by substituting for a the desired endpoint accelerations a_d.

$$\tau = T^{-1}M(q)J^{-1}(q)\left\{a_d - \dot{J}(q,u)u\right\} - T^{-1}\left\{c(q,u) + f(q,u)_{external}\right\}$$

The Jacobian is generally invertable except where the manipulator approaches a singular configuration. For a redundant manipulator, the solution is not unique, and further constraints or objectives can be imposed upon the system to uniquely determine the required torques to cause the desired tip accelerations.

5.2.1 System Momentum Control

The momentum of a system of bodies is a well defined quantity that is easily calculated [15]. Being a function of velocity, mass and moment of inertia of the system, the calculation of momentum (both linear and angular) does not change with kinematic constraints. In the case of a free-flying robot, the total momentum is constant or conserved while floating freely in space, but is not conserved while connected to the space station [1]. The idea behind system momentum control is to cause a multi-body system to follow a momentum trajectory

[1] It is assumed that the space station is so massive that it appears to the robot as inertially fixed.

until a final desired momentum state is reached. At that point the system changes from a closed-chain configuration to an open-chain configuration (releasing from the space station) with the desired final momentum. Momentum control differs from traditional robotic control policies in that instead of controlling one or more state variables (position or velocity of a joint or tip), it forces the momentum of the entire system to follow a commanded momentum trajectory as described in section 4.3.

The controller consists of two parts. The first part combines the dynamical relationship between desired momentum and the generalized accelerations ($\ddot{u}$ terms) with the closed-kinematic-chain constraints. This is analogous to the servo part of the computed torque controller. Having solved for a consistent set of generalized accelerations, the joint torques are then computed.

Momentum Error Equation Given a momentum trajectory as outlined in Chapter 4, an error equation can be written as:

$$\dot{L}_x = \dot{L}_{x_d} + K_{M_x}(L_{x_d} - L_x) + K_{I_x}\int(L_{x_d} - L_x)dt \tag{5.12}$$

$$\dot{L}_y = \dot{L}_{y_d} + K_{M_y}(L_{y_d} - L_y) + K_{I_y}\int(L_{y_d} - L_y)dt$$

$$\dot{H}_z^{\mathrm{CM}} = \dot{H}_{z_d}^{\mathrm{CM}} + K_{M_z}(H_{z_d}^{\mathrm{CM}} - H_z^{\mathrm{CM}}) \tag{5.13}$$

where the subscript "d" denotes desired quantities, L is the linear momentum and H is the angular momentum expressed in the inertial frame. The momentum gains K_M determine the placement of the closed-loop poles, while $\dot{L}_d$ corresponds to the feedforward term. Note that Equations 5.12-5.13 describe type 1 error equations in momentum [24, 9], and cannot take out steady-state errors in position[2]. By adding another integral term to Equation 5.12, the system will exhibit zero steady-state error in position. Steady-state errors in position of the mass center arise when air tubes and wires running through the robot exhibit a constant spring force on the arm joints. Equation 5.13 does not have an integral term, because angular momentum gives rise to nonholonomic constraint equations. Nonholonomic equations require path dependent integration, and are not analytically integrable. Since orientation can be controlled independently of angular momentum, the total linear momentum of the system was controlled to be zero.

Resolving Generalized Accelerations Having determined the commanded momentum, the next step is to resolve the generalized accelerations. This is accomplished by

[2]The integral of linear momentum can also be expressed as the total mass times the position of the mass center.

substituting Equation 3.15 into Equation 3.29 and solving for $\dot{u}_r$ as follows:

$$\begin{bmatrix} \dot{L} \cdot b_1 - c_0 \\ \dot{L} \cdot b_2 - c_1 \\ \dot{H}_z^{CM} \cdot b_3 - \sum_{i=2}^{7} c_i \end{bmatrix} = \begin{bmatrix} \sum_{r=0}^{2} M_{0r}\dot{u}_r + \sum_{s=3}^{7} M_{0s}\dot{u}_s \\ \sum_{r=0}^{2} M_{1r}\dot{u}_r + \sum_{s=3}^{7} M_{1s}\dot{u}_s \\ \sum_{i=2}^{7}\sum_{r=0}^{2} M_{ir}\dot{u}_r + \sum_{i=2}^{7}\sum_{s=3}^{7} M_{is}\dot{u}_s \end{bmatrix} \tag{5.14}$$

$$\dot{u}_s = A_{sr}\dot{u}_r + b_s$$

Like the previous methods of computed torque, system momentum control is another way of finding or specifying a consistent set of generalized accelerations.

Solving for Joint Torques

Having derived a consistent set of commanded accelerations, the last step in the control law is to resolve the joint torques. It is quite common for the solution to this problem to be overconstrained. This means that there is an infinite set of allowable torques that will give the desired accelerations. To find an unique solution, additional objectives must be imposed on the problem. One objective that is commonly used is to minimize the 2 norm of τ. For autonomous vehicles with limited battery supply, this objective corresponds to minimizing the total power consumed. Other objectives include minimizing the 1 norm or the infinity norm, which correspond to minimizing total current and peak torque respectively. For low order systems, it is very computationally efficient to solve the least squares problem for τ using Equation 3.18.

5.3 Practical Issues With Computed Torque

In the derivation of computed torque controllers, certain assumptions were made that impact the design and performance of the system. The following paragraphs describe the two major limitations: continuous time derivation and knowledge of the entire state.

Sample Rate and Computational Delays In all the derivations for computed torque, it is assumed that the plant and control law are running in continuous time and that there is zero time delay for computation. At first glance, it is unclear what the impact this assumption has on the stability and performance of nonlinear systems. The answer to these questions depends on the type of system, the complexity of the system, the actuator authority, and the available computer power. For example, Uhlik [36] has experimentally shown that for a fixed-base two-link arm with elastic drives, a sample rate of 50 Hz was insufficient to stabilize his system with a desired closed-loop bandwidth of 2 Hz. The

problem stems from the fact that the system needs to be sampled at 30 to 50 times the highest natural frequency to insure stability. Because fixed-base robots do not suffer to the same degree as space robots for weight and power constraints, terrestrial high-performance systems can be built where actuator authority is practically unlimited. For example, in the experiment described in [36], applying maximum torque to the shoulder or elbow motors would permanently deform the springs in the drive-train. Therefore, his system was not actuator limited. This can be contrasted with a space-robot, whose acceleration will be small compared to peak motor torque during a "push-off" maneuver.

For rigid body systems, Wampler [37] asserts that one needs to sample at 30 to 50 times the desired *closed-loop bandwidth*. For a bandwidth of 1 Hz, this amounts to a sample period of 20 to 30 milliseconds. Of course bandwidth is not necessarily the criterion one should use in evaluating control performance. For linear systems, closed-loop bandwidth denotes the ability of the system to track sinusoidal inputs. It can be defined as the input frequency at which the output amplitude is 3 dB below that of the input amplitude. This definition is a bit arbitrary, and one needs to question its relevance in selecting sample rates.

The problem is that the definition of bandwidth assumes a linear system and therefore does not account for nonlinear effects such as actuator saturation. For small-amplitude inputs, the "closed-loop" bandwidth could be much higher than for high-amplitude inputs due to saturation. The need for good disturbance rejection can also drive a design to high closed-loop bandwidth. Yet in space, external disturbances will be small in magnitude and frequency content. Therefore, the issues that drive sample rates are not bandwidth, but actuator authority, desired tasks, and system dynamics. One should sample 30 to 50 times the largest desired/achievable frequency content of the states.

Lack of Knowledge of the Plant and State The second assumption in using computed-torque control is knowledge of the entire state. This requirement is not as much an issue for robotic manipulators, since most of the states are single degree of freedom joints. The joint positions can be accurately measured with optical encoders[3] or RVDT sensors, and rate information can be inferred from these signals or estimated using extended Kalman filters (EKF). The hardest states to measure are the orientation and position of the base body. Global positions are measured in the laboratory using an offboard vision system. In space, inertial navigation systems using laser gyros and accelerometers combined with differential GPS could give centimeter accuracy in base position.

[3] Commercial encoders can measure angles of 77μ radians and angular velocity ranging from 0.07 to 4.0 radians/sec to a few percent accuracy.

5.4 Bang-Bang Control

During free-flight, a bang-bang control law was used to perform large angle slews. This control law is optimal in the sense that it produces the minimum time solution to a linear system with limited actuator authority. The derivation is well known (See Bryson [3]), and only the results are given here. If τ_5 is the torque applied by the momentum wheel motor, then

$$\tau_5 = -\frac{T_{max}}{I}\text{sgn}\left\{\dot{q}_2 + \sqrt{\frac{2|q_2|T_{max}}{I}}\right\}$$

where $-T_{max} \leq \tau_5 \leq T_{max}$ and I is the effective moment of inertia about the system center of mass. To prevent chatter after reaching the final attitude, a simple PID control law was employed to regulate the attitude. In the phase plane, this control law will take the robot from any initial angular velocity and angle to zero angular velocity and angle by applying full torque (either positive or negative) and switching at the appropriate time to full torque in the opposite direction.

5.5 In-Parallel Systems

Consider the two systems shown in Fig 5.2. Both drawings show six-bar linkages that form a closed-kinematic-chain configuration. The major difference is the location of two motors, which in one case is inertially fixed, and in the other body fixed on a free-floating target. Fig 5.2a is called an "in-parallel system", because cutting the chain at points "A" and "B" results in two independent two link manipulators. Schneider [27] has taken advantage of this configuration to design a parallel control algorithm that implements object impedance control. Specifying desired accelerations and measuring the external forces at the tips uniquely determines the applied torques to the motors.

This same method does not work in Fig 5.2b. If the links are cut at points "A" and "B", the resulting Jacobians are singular for the two arms because the torque from a motor cannot exert a force at zero distance, but only at a finite distance away. Therefore a cut cannot occur at a motor. If the links are cut at points "C" and "D", the two shoulder motors are still coupled through the dynamics of the floating base. Systems can be decoupled if the cuts result in topological tree structures whose roots are inertially fixed and whose Jacobians are not singular.

Finally, the position or velocity of an object or a link is not important for locomotion, but rather the position and velocity of the *system* of bodies. For this reason, a different approach will be taken to control kinematically coupled systems of rigid bodies that does

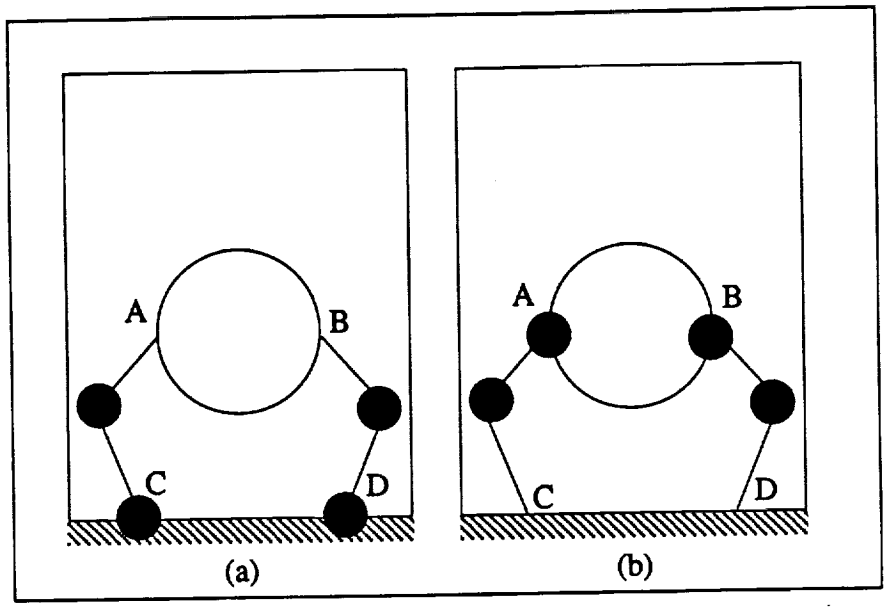

Figure 5.2: **In-Parallel Systems**

The location of actuators determines how easily a closed-kinematic-chain system can be parallelized.

not involve taking advantage of an in-parallel system, but rather incorporates the kinematic constraints with the dynamic equations of motion.

Chapter 6

Experimental Results

In this chapter, experimental results are presented for the momentum controller whose design is described in Chapter 5. These results are intended to verify the design and predicted performance. A comparison is made between simulation and experiment to see how well the model predicted the actual behavior of the robot.

This chapter is divided into two sections: results for an entire LEAP maneuver, and results for joint PD, bang-bang control, and momentum control.

6.1 An Entire LEAP Maneuver

Figure 6.1 is a montage of an actual LEAP maneuver. The robot was commanded to leap straight-back a distance of 3.66 meters, and to complete this task without using thrusters. To accomplish this, the strategic controller guided the robot through the various phases of the maneuver as described below.

Upon receiving the command to leap, along with a desired final momentum state, the robot entered the "Wind-up" phase. After the "Wind-up" phase, the robot accelerated along a straight path, releasing the bar when the velocity of the mass center reached 100 mm/s.

After releasing the bar, the robot brings its arms into a tuck position and rotates 180 degrees . During this period, no attempt is made to control the momentum of the system or to correct for errors in location of the mass center. The angular position of the base is controlled using bang-bang control (minimum time slew) with feedback from an onboard angular rate sensor to a momentum wheel. When rotation is complete, the arms are extended to prepare for landing.

Finally, when both arms grasp the bar, the robot is commanded to stop by implementing

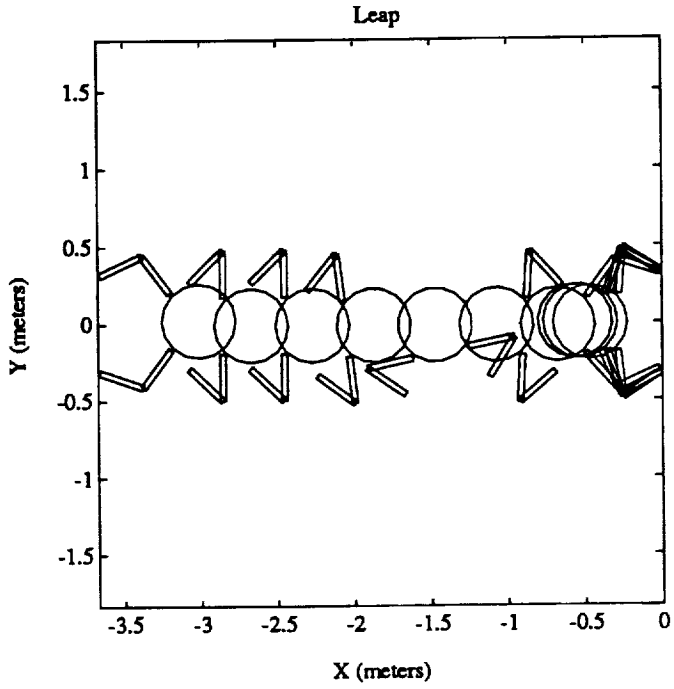

Figure 6.1: **Coordinated Leap Maneuver**

The figure shows the Pushoff, Rotation, and Catch phases for a straight back leap. The robot is going from right to left.

an error equation in momentum only. When the momentum is below a specified threshold, the robot returns to its ready position for another task. With the current experimental setup, the robot can position itself to within 70 mm over a 3.66 m distance without the need for midcourse correction.

Various other angles of departure, between 0 deg and 15 deg were tried. In each case, the robot landed to within 70 mm of the desired location. Limitations in the experimental setup prevented executing launch angles beyond 15 deg. Due to large asymmetries in off-axis pushoffs, one arm imparts the majority of the momentum to the system. With the current link lengths and peak torque capabilities of the motors, a release velocity of 100 mm/s or greater is not possible for departure angles beyond 20 deg. For slower release velocities, grounding effects in the air bearing dominate the experimental results. Of course leaping is but one form of thrusterless locomotion. To translate sideways, crawling might be a preferred method.

6.2 Joint PD and Bang-Bang Control

Although this research focuses on combining different control laws to perform a high level task, each subtask needs to be experimentally verified before the system as a whole can be evaluated. Figure 6.2 shows a plot of the right-shoulder and right-elbow joints going through commanded 73 degree and 95 degree slews. This maneuver is identical to an arm tuck, which the robot performs to minimize its moment of inertia. The control law is a simple PD error law wrapped around each joint.

Although it insures stability, the PD control law is not designed to give high performance for a nonlinear system. No attempt was made to model the system or to compensate for steady state errors. The steady state errors can clearly be seen in Figure 6.2 as the difference between desired position (the dashed lines) and the measured position. A PID control law was also designed which eliminated the steady state error. Since the only function of this control law was to move the arms into a tuck position, great accuracy in position and trajectory following was not necessary, and the performance was deemed adequate.

The desired trajectories are formulated as fifth-order polynomials to ensure smooth transitions in angular acceleration as well as in angular velocity and position.

$$\begin{aligned}
\theta_d(t) &= a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \\
\dot{\theta}_d(t) &= 5a_5t^4 + 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1 \\
\ddot{\theta}_d(t) &= 20a_5t^3 + 12a_4t^2 + 6a_3t + 2a_2
\end{aligned}$$

If the initial and final angles are θ_i and θ_f, and t_f is a desired elapsed slew time, then the six unknown coefficients a_i are given by:

$$\begin{aligned}
a_0 &= \theta_i \\
a_1 &= \dot{\theta}_i \\
a_2 &= \frac{\ddot{\theta}_i}{2} \\
a_3 &= \frac{(\ddot{\theta}_f - 3\ddot{\theta}_i)t_f^2 - (8\dot{\theta}_f + 12\dot{\theta}_i)t_f + 20(\theta_f - \theta_i)}{2t_f^3} \\
a_4 &= \frac{-(2\ddot{\theta}_f - 3\ddot{\theta}_i)t_f^2 + (14\dot{\theta}_f + 16\dot{\theta}_i)t_f - 30(\theta_f - \theta_i)}{2t_f^4} \\
a_5 &= \frac{(\ddot{\theta}_f - \ddot{\theta}_i)t_f^2 - 6(\dot{\theta}_f + \dot{\theta}_i)t_f + 12(\theta_f - \theta_i)}{2t_f^5}
\end{aligned}$$

It is important to specify acceleration profiles on lightweight arms to prevent the jerkyness

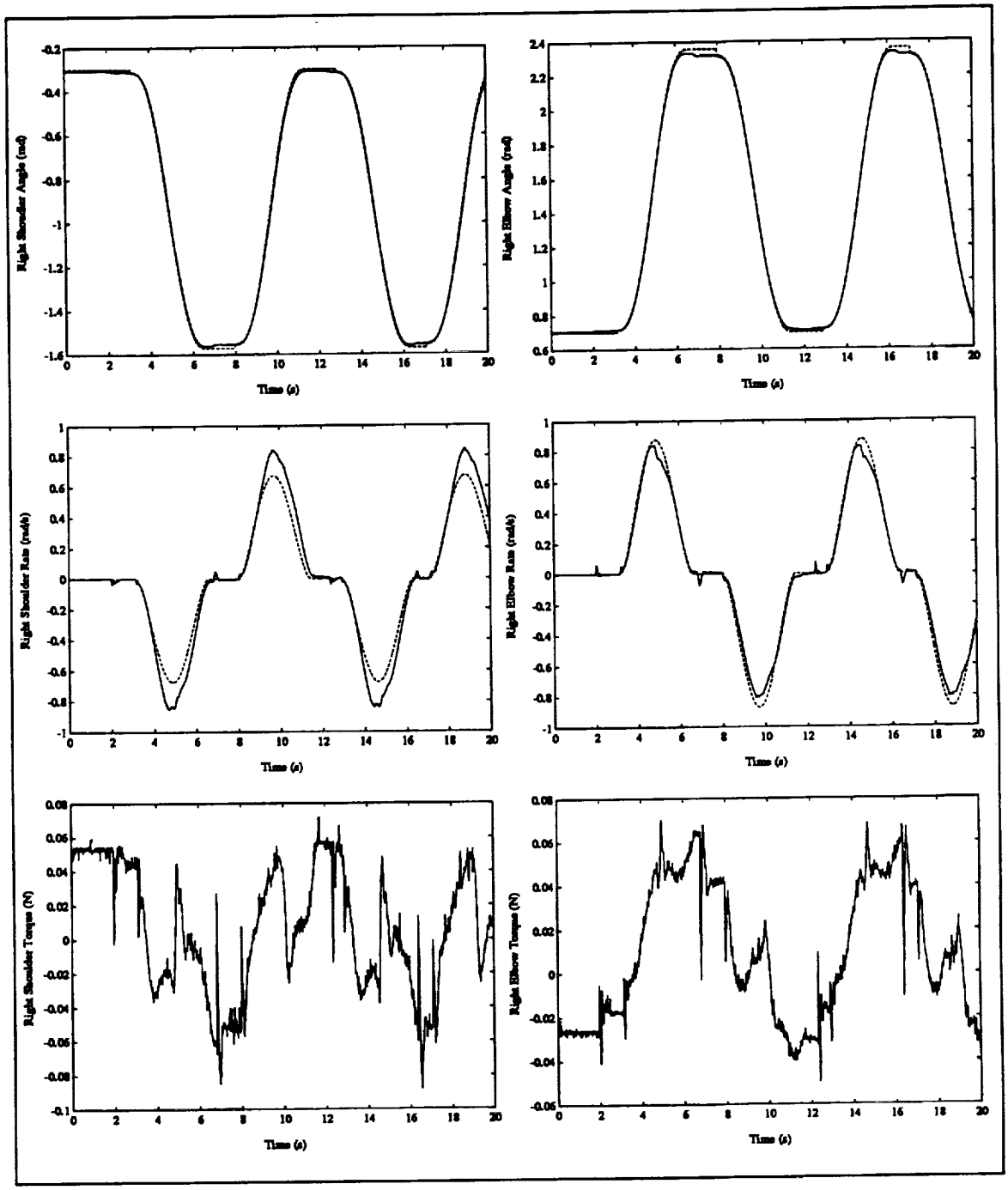

Figure 6.2: **Nominal Slews of a PD Controller for a Two Link Arm**

A series of 3.5 second slews were commanded to each joint separately. The dashed lines represent desired values, while the solid lines show experimentally measured data. While position tracking is good, velocity tracking is rather poor. Coupling between joint results in large variations in torque (bottom plots).

associated with large changes in acceleration. Actually, humans tend to follow minimum jerk[1] trajectories while performing pick and place tasks [8].

The base orientation was controlled by a different method: the time-optimal bang-bang controller. Both angular position and angular velocity of the base were used to determine whether the momentum wheel motor should supply maximum positive or maximum negative torque. Figure 6.3 shows the orientation and angular velocity for a 180 deg slew. During the slew, the arms were regulated about a 45 deg angle. The degradation in the measurement results from chatter while the controller is regulating about zero. For a system with a small T_{max}/I (See section 5.4), the peak accelerations are small; inertia acts like an acceleration filter. Therefore, the system does not experience jerky motions between maximum changes in input torque. The T_{max}/I ratio is 0.164 rad/sec^2 in the nominal tuck configuration. In an actual leap maneuver, the bang-bang control law is switched off after reaching the desired angle, and a simple PD controller is installed to regulate the final orientation of the base.

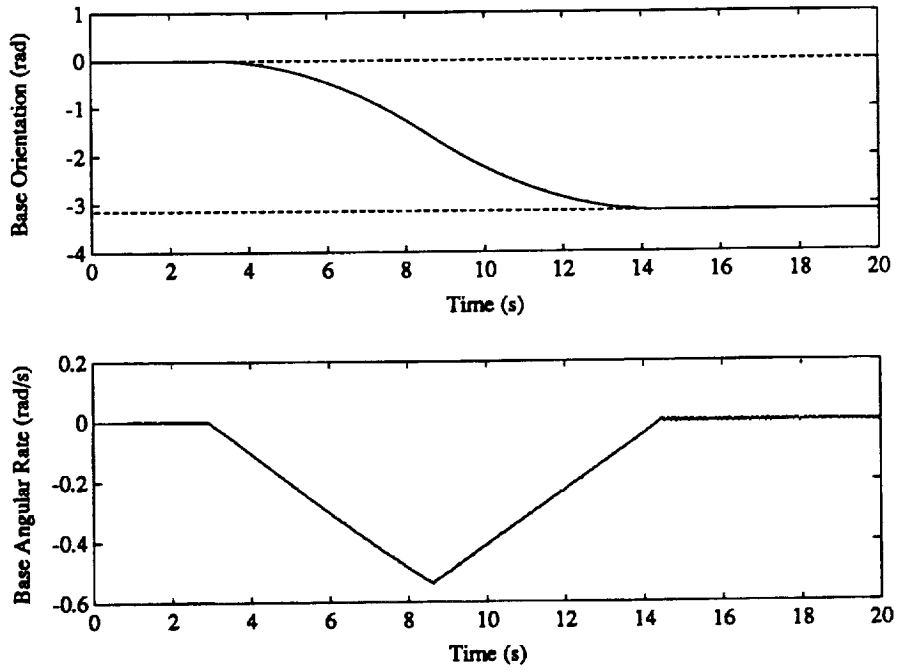

Figure 6.3: **Orientation and Angular Velocity of the Base for Minimum Time Slew**

[1] Jerk is the derivative of acceleration.

6.3 Momentum Control

Figure 6.4 shows the response of the system to a step command in the position of the mass center. This test was performed to experimentally verify that the feedback part of the control law caused the system to behave like a second order system with damping ratio $\zeta = 0.8$ and natural frequency $\omega = 1.7$ rad/sec. The slight deviation between experimental and predicted behavior is due to unmodelled friction effects in the air-tubing and wiring that run through the arms. Figure 6.5 shows a comparison between desired, simulated and

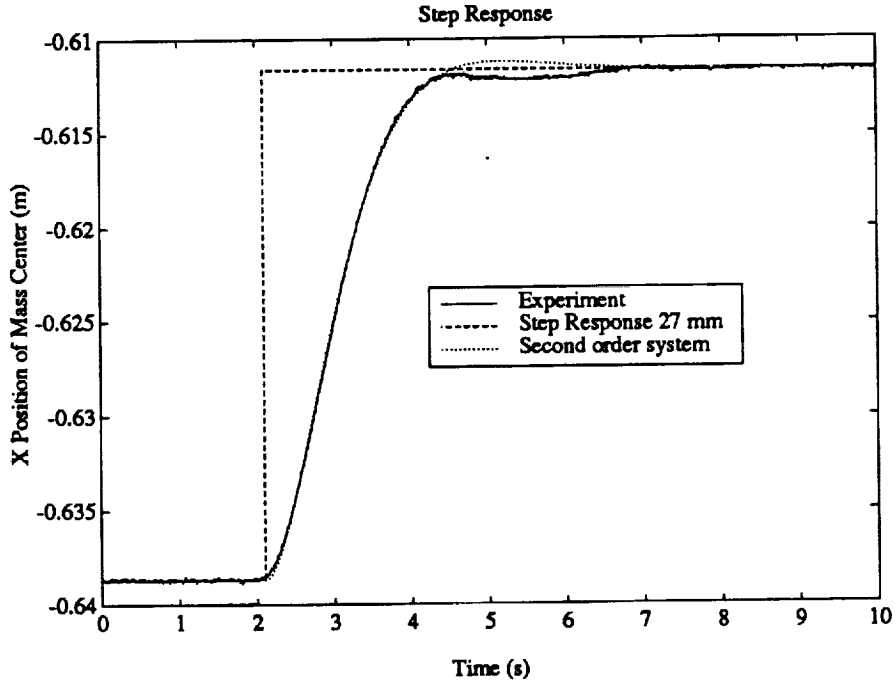

Figure 6.4: **Response To Step Input**

Plot of the center-of-mass position to a commanded step input. Also plotted is the ideal response of a second order system to a step input. This plot shows excellent agreement between the theoretical and actual error-equation pole positions (dashed vs. solid line). This plot demonstrates that under momentum control, the system will respond to errors in trajectory tracking like a second-order system.

experimentally measured positions and velocities for the mass center during the windup and pushoff phases. The steady-state error in position at point B, the end of the windup, is due to the air tubes applying a torque to the elbow joints. Since there is no integral error in position, the result was a small steady state error. However, no such error occured in

momentum, since there is integral error in momentum, and the system tracked the desired momentum trajectory well. At point C, the robot released the bar and floated to the other side of the table.

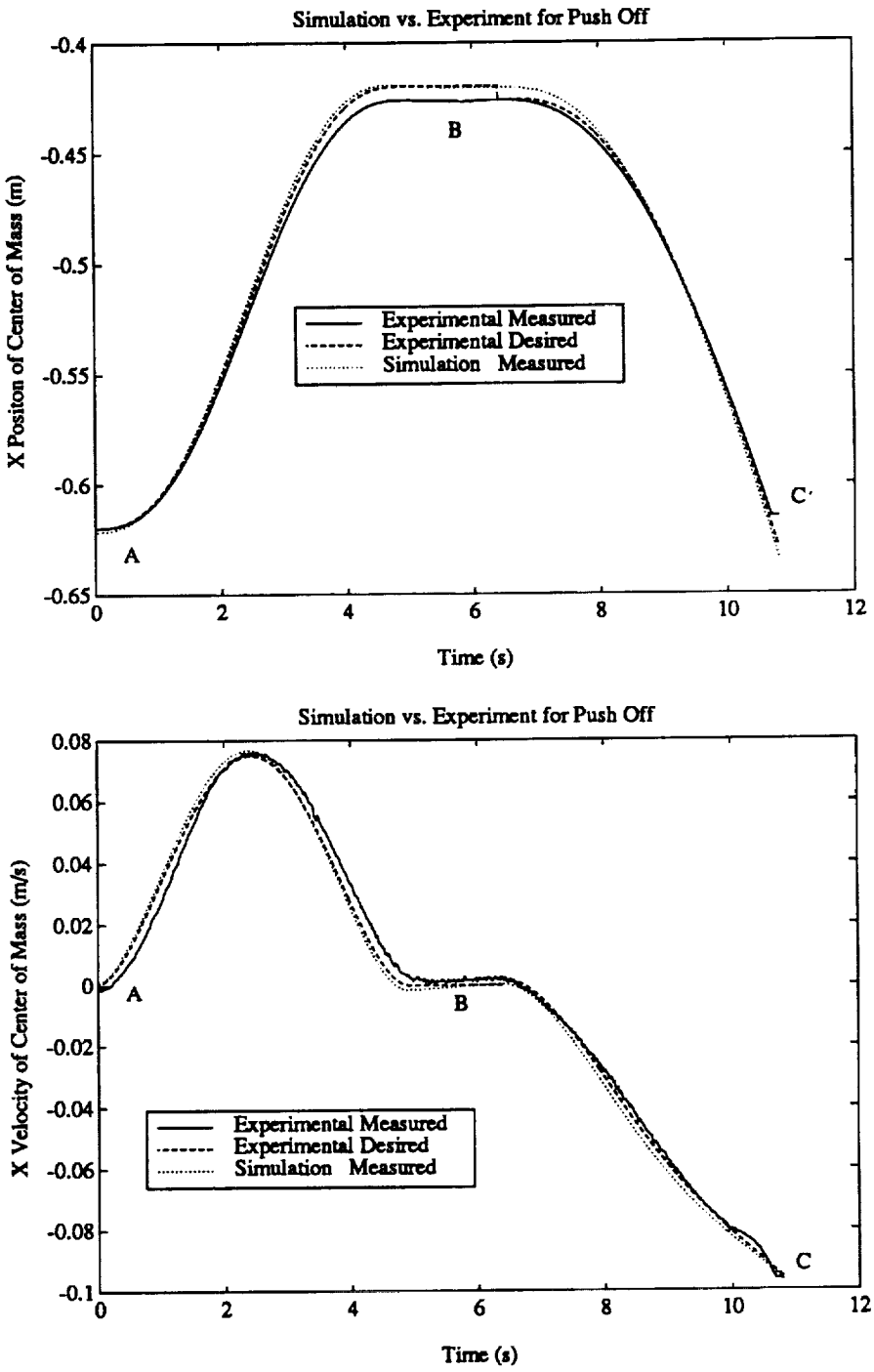

Figure 6.5: **Experimental vs. Simulation**

The position and velocity of the mass center of the robot is plotted during the pushoff phase.

Chapter 7

Extensions to Three Dimensions

Thus far, most of the discussion has focused upon locomotion in two dimensions. Although laboratory experiments were performed in two dimensions, many of the ideas easily extend to three. The derivation of the equations of motion, the design of the strategic controller and the momentum controller readily extend to three dimensions, and require no modification to the theory. Issues that do need to be addressed involve orientation and vehicle design.

This chapter is divided into three sections. The first section deals with an abstract treatment of orientations of deformable bodies. By using purely geometric arguments, the net rotation can be calculated from a set of path dependent motions. The second section describes a method of performing attitude control using a fifth-order trajectory in Euler parameters and resolved acceleration or computed torque. The last section describes some design considerations when going from two to three dimensions.

7.1 Orientation in 3 Dimensions

Although the total angular and linear momentum of an object is constant in the absence of external forces and torques, the orientation of that object need not be constant. Studies have been done [17, 14] which demonstrate arbitrary orientation with zero angular momentum. A more general treatment of the subject was done by Shappere and Wilczek [30], which presented a kinematic formulation of this problem in terms of a gauge structure over the space of shapes that a body may assume. Their work is presented here along with some comments on how it might be applied to three dimensional robotic design.

Consider the set of all possible shapes of a deformable body, which we will call configuration space. Each point in this space corresponds to a unique shape with orientation and position of the mass center. This set can be partitioned into equivalence classes by an

equivalence relation; shapes that differ only in center of mass location and orientation are equivalent. When no external forces act on a body, the center of mass is unchanged for every shape in configuration space. Thus we will work in the center of mass frame, where the space of located shapes is just the space of unlocated shapes *with orientation.*

An example will help to illustrate some of these ideas. Consider a deformable body consisting of a hinge, separated by an angle θ, where $0 \leq \theta < \pi$. An equivalence class of unlocated shapes is the set of all shapes with the same angle θ as shown in Figure 7.1. The question we wish to answer is: what is the net rotation or change in orientation that

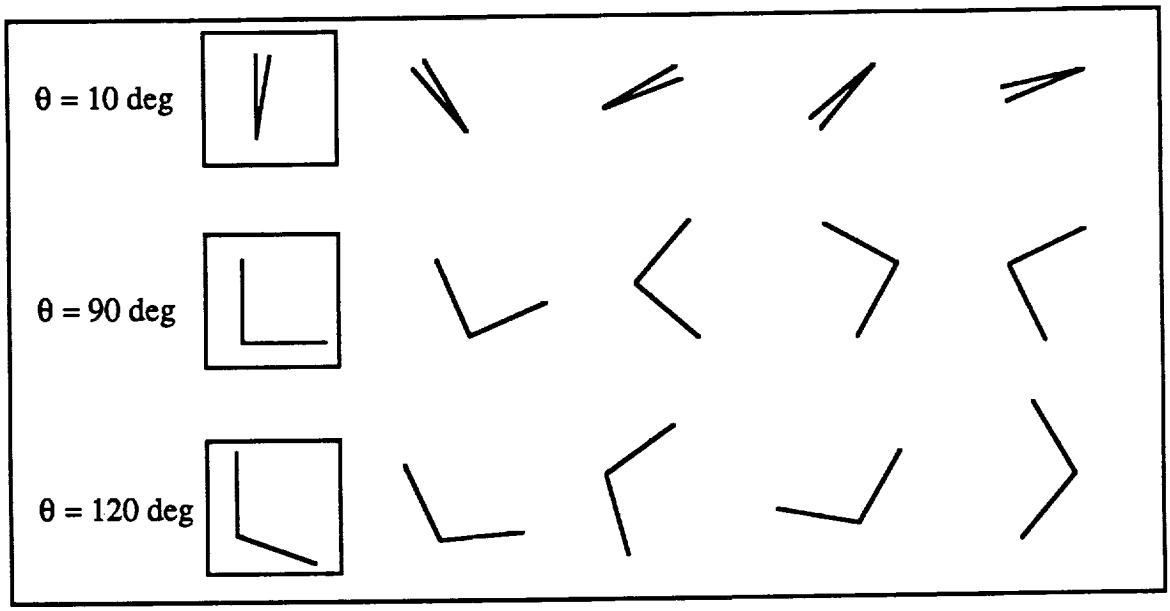

Figure 7.1: **Equivalence Class of Unlocated Shapes**

The set of possible shapes can be partitioned into equivalence classes of shapes that differ only in orientation. The shapes in the boxes were chosen to be the standard shapes for that class.

results when a deformable body goes through a given sequence of unoriented shapes in the absence of external forces or torques? The net rotation can be computed by making use of the law of conservation of angular momentum. This nonholonomic constraint will be sufficient to determine fully the net rotation of a deformable body.

Suppose one has chosen a set of standard body-fixed reference axes for each possible unoriented shape. This is accomplished by choosing one shape from each equivalence class and assigning an arbitrary dextral set of x, y, z axes. Therefore, a standard shape is one point of the equivalence class of located shapes corresponding to an unlocated shape. In our example, the shapes inclosed in a square box comprise the set of standard shapes. For a

given sequence of these standard shapes $S_o(t)$, we wish to find the corresponding sequence of physically oriented shapes $S(t)$, which are related by the rotation matrix $R(t)$:

$$S(t) = R(t)S_o(t) \tag{7.1}$$

where $R(t)$ is a 3×3 rotation (direction cosine) matrix which depends on the reference axes for $S_o(t)$. Of course the path of physical shapes $S(t)$ should be invariant to the arbitrary choice of standard shapes. If one were to choose a different set of standard shapes:

$$\tilde{S}_o = \Omega[S_o]S_o \tag{7.2}$$

then the new rotation matrix would be:

$$\tilde{R}(t) = R(t)\Omega^{-1}[S_o(t)] \tag{7.3}$$

to insure that $S(t)$ is unchanged. One can write a differential equation in $R(t)$ by defining:

$$\frac{dR}{dt} = R\left[R^{-1}\frac{dR}{dt}\right] \equiv RA \tag{7.4}$$

$$R(t) = \exp\int_0^t A(\tau)d\tau \tag{7.5}$$

The matrix A can be thought of as an operator on R which maps an infinitesimal deformation of $S_o(t)$ into an infinitesimal rotation. It can be shown that A, also known as the *angular velocity matrix* is skew symmetric, meaning that only 3 of the nine elements are independent. This fact will become very useful in subsequent calculations with A. Equation 7.5 is actually invariant under arbitrary time rescalings, suggesting that one should be able to write Equation 7.5 in a completely geometric form. To see why this is so, consider a mapping $t \to f(t)$. Then $dt \to \dot{f}dt$ while $A \to A/\dot{f}$ since:

$$A(f(t)) = R^{-1}(f(t))\frac{dR(f(t))}{dt} = R^{-1}(f(t))\frac{dR(f(t))}{df(t)}\dot{f}$$

$$A(f)/\dot{f} = R^{-1}(f)\frac{dR(f)}{df}$$

Dynamics of a free body The dynamics of a body in the absence of any external forces or torques is completely determined by the conservation of momentum laws. Without loss of generality, consider the total angular momentum to be zero.

If one considers a body consisting of a collection of N point masses $m^1 \ldots m^N$ at positions $x^1 \ldots x^N$, then the total angular momentum of the system is:

$$H_i = \epsilon_{ijk}\sum_{n=1}^{N} m^{(n)}x_j^{(n)}\dot{x}_k^{(n)} \tag{7.6}$$

where ϵ_{ijk} is the permutation symbol in three dimensions with i, j, k taking on values 1, 2 or 3, and:

$\epsilon_{ijk} = 0$ if two suffixes are equal.

$\epsilon_{ijk} = 1$ if ijk is an even permutation of the sequence 123.

$\epsilon_{ijk} = -1$ if ijk is an odd permutation of the sequence 123.

An identity which is useful for tensor cross products in three dimensions is [34]

$$\epsilon_{mrs}\epsilon_{mpq} = \delta_{rp}\delta_{sq} - \delta_{rq}\delta_{sp} \tag{7.7}$$

where δ_{ij} is the Kroniker delta taking the value 1 when $i = j$ and 0 otherwise.

Recall that the *unlocated* shapes are related to the oriented shapes at time t by the relation:

$$x^{(n)}(t) = R(t)\tilde{x}^{(n)}(t) \tag{7.8}$$

where $R(t)$ is a 3×3 rotation matrix. Substituting Equation 7.8 into Equation 7.6 gives an expression of the angular momentum in terms of $\tilde{x}^{(n)}$ and $R(t)$.

$$H_i = \epsilon_{ijk} \sum_{n=1}^{N} m^{(n)} \left[R_{jl}\tilde{x}_l^{(n)} R_{km}\dot{\tilde{x}}_m^{(n)} + R_{jl}\tilde{x}_l^{(n)} \dot{R}_{km}\tilde{x}_m^{(n)} \right] \tag{7.9}$$

By setting $H_i = 0$ in Equation 7.9, multiplying by R^{-2} and performing some algebra, one obtains:

$$A(t)_{ij} = (R^{-1}\dot{R})_{ij} = -\epsilon_{ijk}\omega_k = \epsilon_{ijk}\tilde{I}_{kl}^{-1}\tilde{H}_l \tag{7.10}$$

where $\tilde{I}$ is the inertia tensor of standard shapes $S_o(t)$, and $\tilde{L}$ is the apparent angular momentum of $S_o(t)$ at time t:

$$\tilde{I}_{kl} \triangleq \sum_{n=1}^{N} m^{(n)} \left((\tilde{x}^{(n)})^2 \delta_{kl} - \tilde{x}_k^{(n)}\tilde{x}_l^{(n)} \right) \tag{7.11}$$

$$\tilde{H}_l \triangleq \epsilon_{ljk} \sum_{n=1}^{N} m^{(n)} \tilde{x}_j^{(n)} \dot{\tilde{x}}_k^{(n)} \tag{7.12}$$

where use has been made of Equation 7.7 and the fact that $A(t)$ is skew symmetric. Equations 7.10 and 7.5 provide a complete solution to the problem of determining net rotation of a deformable body.

For a multi-body space robot, the calculation of A can be computed in a more convenient form by making use of Equation 3.24. Notice that this equation is linear in u_i, the generalized speeds. If one orders the u_i such that the first three generalized speeds correspond to the measure numbers of the angular velocity vector given in Equation 7.24, then

$$H_i = \mathcal{I}_{ij}\omega_j + \tilde{I}_{in}u_n$$

where $i,j = 1,2,3$ and $n = 4 \ldots N$ where N is the number of degrees of freedom of the system. If the angular momentum is zero, then

$$\omega_i = \mathcal{I}_{im}^{-1} \tilde{I}_{mn} u_n \tag{7.13}$$

$$A(t) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{7.14}$$

7.2 Attitude Control in 3 Dimensions

In spacecraft attitude control, the equations of motion are usually linearized about a fixed desired orientation. This is a very good assumption while regulating about a fixed attitude, and the literature is full of examples on how to implement 3-axis attitude control. The problem becomes more complicated when performing rapid slews from one orientation to another. Typically, the satellite is rotated about each principal axis independently, until the desired final orientation is achieved. In this way both the kinematics and the dynamics are decoupled into a set of *linear* equations. This technique presupposes that the torque vector is aligned with a principal axis of the spacecraft and that all other cross terms in the angular velocity vector are zero.

In orienting a space robot, the problem is complicated by the fact that the principal axes change with configuration. Therefore, one should not expect momentum wheels or other torque devices to align with any particular geometry of the robot. Even if the torque vectors were aligned with the principle axes, one would like to specify an initial and final orientation, and have the robot reorient itself in one smooth continuous maneuver. One method proposed by Dwyer [7] transforms the equations of motion to exhibit linear input-output behavior. The following section outlines a scheme for general 3 dimensional attitude control of a free-flying robot. By combining this method with system momentum control, the user has an unified approach to thrusterless space robotic locomotion in three dimensions.

Consider the problem of rotating a rigid body from an initial orientation O_i to a final orientation O_f. A theorem due to Euler on rotation states that every change in relative orientation of two rigid bodies can be produced by means of a simple rotation [16]. Let λ be a unit vector parallel to the axis of rotation, and θ be the angle of rotation. Then four scalar quantities, $\epsilon_1, \ldots, \epsilon_4$, called *Euler parameters*[1], can be defined as:

$$\epsilon \triangleq \lambda \sin \frac{\theta}{2} \tag{7.15}$$

[1]The Euler Parameters should not be confused with the permutation symbol ϵ_{ijk} of Section 7.1.

$$\epsilon_i \triangleq \boldsymbol{\epsilon} \cdot \boldsymbol{a}_i = \boldsymbol{\epsilon} \cdot \boldsymbol{b}_i \qquad (i = 1,2,3) \tag{7.16}$$

$$\epsilon_4 \triangleq \cos\frac{\theta}{2} \tag{7.17}$$

where $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3$ is a dextral set of orthogonal unit vectors fixed in inertial space, with $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ fixed in the body reference frame. The Euler parameters are not independent, but must satisfy the condition:

$$\epsilon_4^2 = 1 - \epsilon_1^2 - \epsilon_2^2 - \epsilon_3^2 \tag{7.18}$$

The Euler parameters can also be expressed in terms of direction cosines as follows:

$$C_{ij} \triangleq \boldsymbol{a}_i \cdot \boldsymbol{b}_j \qquad (i,j = 1,2,3) \tag{7.19}$$

$$\epsilon_4 = \frac{1}{2}(1 + C_{11} + C_{22} + C_{33})^{\frac{1}{2}} \tag{7.20}$$

$$\epsilon_1 = \frac{C_{32} - C_{23}}{4\epsilon_4} \tag{7.21}$$

$$\epsilon_2 = \frac{C_{13} - C_{31}}{4\epsilon_4} \tag{7.22}$$

$$\epsilon_3 = \frac{C_{21} - C_{12}}{4\epsilon_4} \tag{7.23}$$

If one defines the angular velocity vector ${}^A\boldsymbol{\omega}^B$ of B relative to A as:

$${}^A\boldsymbol{\omega}^B \triangleq \omega_1 \boldsymbol{b}_1 + \omega_2 \boldsymbol{b}_2 + \omega_3 \boldsymbol{b}_3 \tag{7.24}$$

then the relation between the measure numbers ω_i of angular velocity and the derivatives of the Euler parameters can be given as:

$$\epsilon \triangleq [\ \epsilon_1 \quad \epsilon_2 \quad \epsilon_3 \quad \epsilon_4\ ]^T \tag{7.25}$$

$$\omega \triangleq [\ \omega_1 \quad \omega_2 \quad \omega_3 \quad 0\ ]^T \tag{7.26}$$

and

$$E = \begin{bmatrix} \epsilon_4 & -\epsilon_3 & \epsilon_2 & \epsilon_1 \\ \epsilon_3 & \epsilon_4 & -\epsilon_1 & \epsilon_2 \\ -\epsilon_2 & \epsilon_1 & \epsilon_4 & \epsilon_3 \\ -\epsilon_1 & -\epsilon_2 & -\epsilon_3 & \epsilon_4 \end{bmatrix} \tag{7.27}$$

$$\dot{\epsilon} = \frac{1}{2} E\omega \tag{7.28}$$

$$\omega = 2E^T\dot{\epsilon} \tag{7.29}$$

The derivations for the above equations can be found in [16]. Two other quantities of interest are the derivatives of Equations 7.28 and 7.29 which will be needed for control. They are:

$$\ddot{\epsilon} = \frac{1}{2}(\dot{E}\omega + E\dot{\omega}) \tag{7.30}$$

$$\dot{\omega} = 2(\dot{E}^T\dot{\epsilon} + E^T\ddot{\epsilon}) \tag{7.31}$$

The last equation can be written in a slightly more compact form as:

$$\begin{bmatrix} \dot{\omega_1} \\ \dot{\omega_2} \\ \dot{\omega_3} \end{bmatrix} = 2 \begin{bmatrix} \epsilon_4 & \epsilon_3 & -\epsilon_2 & -\epsilon_1 \\ -\epsilon_3 & \epsilon_4 & \epsilon_1 & -\epsilon_2 \\ \epsilon_2 & -\epsilon_1 & \epsilon_4 & -\epsilon_3 \end{bmatrix} \begin{bmatrix} \ddot{\epsilon_1} \\ \ddot{\epsilon_2} \\ \ddot{\epsilon_3} \\ \ddot{\epsilon_4} \end{bmatrix} \tag{7.32}$$

7.2.1 Desired Trajectories in Euler Parameters

Consider the task of reorienting a rigid body from an initial orientation and angular velocity to a final orientation and angular velocity at some time t_f later. One could easily cause the path to follow a fifth order trajectory such as:

$$\epsilon_{i_d}(t) = a_{i5}t^5 + a_{i4}t^4 + a_{i3}t^3 + a_{i2}t^2 + a_{i1}t + a_{i0} \qquad (i = 1, 2, 3) \tag{7.33}$$

where the a_{ij} would be determined by the initial and final conditions as described in Appendix 4.3. The fourth Euler Parameter, ϵ_4 can be solved in terms of the other three parameters as follows:

$$\epsilon_4 = \sqrt{1 - \epsilon_1^2 - \epsilon_2^2 - \epsilon_3^2} \tag{7.34}$$

$$\dot{\epsilon_4} = -(\epsilon_1\dot{\epsilon_1} + \epsilon_2\dot{\epsilon_2} + \epsilon_3\dot{\epsilon_3})/\epsilon_4 \tag{7.35}$$

$$\ddot{\epsilon_4} = -(\epsilon_1\ddot{\epsilon_1} + \epsilon_2\ddot{\epsilon_2} + \epsilon_3\ddot{\epsilon_3} + \dot{\epsilon_1}^2 + \dot{\epsilon_2}^2 + \dot{\epsilon_3}^2 + \dot{\epsilon_4}^2)/\epsilon_4 \tag{7.36}$$

where $\epsilon_4 \neq 0$. Should $\epsilon_4 = 0$ and $\dot{\epsilon_4} \neq 0$, then[2]:

$$\dot{\epsilon_4}^2 = -(\epsilon_1\ddot{\epsilon_1} + \epsilon_2\ddot{\epsilon_2} + \epsilon_3\ddot{\epsilon_3} + \dot{\epsilon_1}^2 + \dot{\epsilon_2}^2 + \dot{\epsilon_3}^2)$$

$$\ddot{\epsilon_4} = -(3(\dot{\epsilon_1}\ddot{\epsilon_1} + \dot{\epsilon_2}\ddot{\epsilon_2} + \dot{\epsilon_3}\ddot{\epsilon_3}) + \epsilon_1\epsilon_1^{(3)} + \epsilon_2\epsilon_2^{(3)} + \epsilon_3\epsilon_3^{(3)})/3\dot{\epsilon_4}$$

7.2.2 Error Equation in Euler Parameters

Similar to the error equation in momentum, an error equation can be written in terms of the Euler Parameters. By substituting the values of ϵ and ω into Equation 7.28, the error

[2]This is the case when the *Rodrigues parameters* are infinite. The parameter ϵ_4 is zero when rotating through an angle of π radians about an arbitrary line.

equations becomes:

$$\ddot{\epsilon}_{i_c} = \ddot{\epsilon}_{i_d} + K_{v_i}(\dot{\epsilon}_{i_d} - \dot{\epsilon}_i) + K_{p_i}(\epsilon_{i_d} - \epsilon_i) \qquad (i = 1,2,3) \tag{7.37}$$

where the subscript c denotes commanded input to the controller and the subscript d denotes the desired value of the Euler parameter. Using the values of $\ddot{\epsilon}_{i_c}$ and Equation 7.32, one can calculate $\dot{\omega}$. By formulating the dynamical equations of motion in such a way that the set of generalized velocities includes the angular velocity terms ω, (e.g. $u_i = \omega_i$ i = 1, 2, 3) one can solve for the torques by the computed torque method described in Chapter 5.

7.3 Design Criteria In Three Dimensions

The experiments conducted in this research were carried out in two dimensions. Obviously, space robots that work and move in space will have to be designed to operate in three dimensions. This section discusses some of the issues involved in operating a free-flying space robot in three dimensions.

An important issue is the number of arms and degrees of freedom per arm that should be incorporated into a space robot. Manipulating an object requires at least six DOF (degrees of freedom). If multiple arms cooperate, then each arm is not required to possess six DOF. For example, in the plane, two two-link SCARA arms can manipulate a 3 DOF floating target. Redundancy can be put to good use only at the cost of increased mechanical and computational complexity. Also, exerting large torques at the tips of the manipulators requires high gearing and large motors. This problem can be reduced by using the mechanical advantage available with multiple arms. Using a pair of four DOF arms along with a simple three DOF arm would constitute an eighteen DOF robot. Although controlling an 18 DOF system seems very difficult with existing computers, many tasks could decouple the control problem. For example, if one arm was attached to the space station, its motors along with three reaction wheels could control attitude while the other arms manipulated a target. The motion of the arms could be treated as disturbances on attitude control, thus decoupling the two control systems. This is equivalent to the work done in controlling mini-manipulators on a flexible base [19].

In addition to manipulating an object, the arms can be used to relocate/reorient the robot. It can be shown that one arm possessing six degrees of freedom is sufficient to perform a "pushoff" or "catch" task while connected to a structure through a ball joint [3]. Angular momentum about the ball joint is constant, since no external forces or torques are imparted on the vehicle about that point. However, forces are transmitted through the ball

[3]The idea of using only one arm for locomotion was suggested to the author by Prof. DeBra of Stanford University.

joint, so that upon release, one can achieve a desired momentum state. One disadvantage of this approach is that large joint torques may have to be applied by the arm to bring the system to rest.

The two-arm approach demonstrated in this thesis easily generalizes to three dimensions. With two two-link arms, the robot can control its momentum in a plane. With the aid of one more momentum wheel to initially align that plane in a desired direction, the robot can effectively control linear momentum in three dimensions. Angular momentum control in three dimensions requires some modifications to the arms, since a screw motion is not possible with planar SCARA arms.

Locomotion tasks are easier to perform in space than on earth. In the absence of gravity, the robot does not become unstable when one or more arms (legs) release from the bar. Using three legs to crawl along a truss structure can be easily accomplished by using a combination of endpoint control to reposition the tips and momentum control to move the mass center. Leaping from one point to another can be done in three dimensions as was demonstrated in two dimensions. The only concern is what would happen if the robot missed the destination. One possibility is to have a tether attached to the space structure at the point of release. If the maneuver was successful, a computer onboard the space station could release the lock, and the robot could reel in the tether. Otherwise, the robot could reel itself back to the original starting point and try again. Another option is to use thrusters for midcourse corrections.

Chapter 8

Conclusions

This Chapter is divided into two sections: A summary of the research presented in this dissertation, and recommendations for future research. Some of the generic contributions of the work are noted in Chapter One.

8.1 Summary

This thesis comprised an experimental study of the issues involved in thrusterless locomotion for space robotic systems. Two major topics were addressed: design and implementation of dynamic controllers, and use of strategic control to carry out task level commands.

Momentum Control A new method for controlling the dynamic behavior of a rigid-body system was developed. Called *system momentum control*, this method allows one to specify momentum trajectories instead of Cartesian-space or joint-space trajectories. This was shown to be very useful in controlling systems whose plant dynamics changed abruptly due to kinematic constraints. The feedforward portion of the controller allows for smooth trajectory following and improved performance, while the feedback portion allows errors to decay exponentially. This approach is amenable to systems with limited actuator authority that could saturate with large feedback gains.

Current space robotic design and research has focused primarily on issues of object manipulation. Thus, one assumes that the objects are in the workspace of the manipulators. For fixed automation in a structured environment, this assumption may not constrain the effectiveness of the robotic system. However, in space, a fundamental limitation is the ability to overlap the task's location with the robot's workspace. Designing mobility into the task specification instead of constraining the workspace increases the robot's usefulness. The research presented here focuses on the issues of mobility and locomotion.

A new approach to space robot locomotion was presented which obviates propulsion as the primary means of imparting momentum to the robot: Instead of using thrusters, judicious use is made of the arms to *transfer* momentum to the robot from various structures in the environment. A multi-arm robot can push off or grab onto a structure, thereby imparting momentum to itself. This cooperative two-arm maneuver would be exceedingly difficult to perform by manually controlling the arms because of the nonlinearities in the plant and the time-critical path that must be followed. Leaping from one location to another introduces complexities in dynamics and controller design as the robot's configuration changes from closed-kinematic chains to open-kinematic chains. By focusing on controlling momentum, a configuration-independent quantity, system momentum control allows the robot to leap precisely from one place to another while accounting for nonlinear forces and kinematic constraints.

Combining system momentum control with operational-space control allows for additional capabilities such as crawling. Combining different forms of locomotion such as leaping and crawling reduces that accuracy needed by a single aspect in the overall maneuver. For example, to relocate the robot to within 10 mm over a 50 m leap would require an accuracy in pushoff angle of 0.2 mrad without midcourse correction. If the target area is large enough so that the robot can land 200 mm away and crawl to the desired location, the initial accuracy required is reduced by a factor of 20. This may not always be feasable, but should be considered when designing the robot *and* the environment in which it will work.

Strategic Control A strategic controller allows for a higher level of abstraction in formulating and implementing desired *tasks*, instead of commanding desired *states*. Implemented as a finite-state machine, the strategic controller provides a mechanism for switching between synchronous dynamic controllers based upon asynchronous events. Combining a strategic controller with a dynamic controller proved to be a very powerful method for specifying tasks that switch between various control laws during discrete changes in kinematic constraints. This method is also compatible for manipulation tasks and thus fits into an overall framework for combining the various capabilities of a space robot.

A two-arm free-floating robot was constructed which successfully demonstrated many aspects of thrusterless locomotion. By floating on an air-bearing, the robot simulates the zero-g drag-free environment of space in two dimensions. The addition of a momentum wheel and special end-effectors allows the robot to relocate and reoriented itself, without thrusters, in a precise and accurate manner. This can be used to good advantage in space, where the cost of propellant, a finite resource, will ultimately determine the usefulness of mobile robots. Other benefits from reduced thruster usage include a cleaner environment, higher reliability and reduced cost.

Although many new features were incorporated into the current-generation space robot, some problems still remain. The air gap beneath the vehicle is only 50.8 μm thick, so dust particles from the room cause localized grounding of the base plate. It is very difficult to remove all dust and other small particles from a large surface without moving to a clean room environment. An electrostatic air cleaner was installed to remove particles down to 0.01 micron diameter, but this approach has led to mixed results. Therefore, experiments involving gross motions such as leaping across the table are difficult to perform repeatably.

The current ethernet communication link is via a fiber optic cable. Even with one vehicle, the cable often gets fouled around the vehicle. This situation will be exacerbated when multiple vehicles cooperate together. The obvious solution is to install wireless ethernet transceivers on the robots. This technology will become available in the near future.

Lastly, heavy current usage on the batteries greatly reduces shelf life and increases internal resistance. NiCd batteries perform best when supplying a small steady current. This is not the case during a pushoff maneuver, when peak current is demanded for short periods of time. Two possible solutions to this problem are gearing the motors to increase the effective torque or installing higher power density batteries, such as sodium sulphur batteries.

8.2 Recommendations for Future Research

No piece of research is complete in itself. It is hoped that this work will provide some answers as well as raise new questions to many of the problems associated with space robot locomotion. This section explores some possible extensions for future research.

8.2.1 Enhancements to the Robot

The design and construction of the free-flying robot is a continuing process. Several enhancements should be made to the improve its performance and capabilities. The most important component that should be added is a vision system. This will allow for accurate vehicle position at all times, not just when the robot is in contact with the bar. Also, accurate endpoint position information will allow object manipulation tasks to be incorporated with leaping and crawling.

The difficulties mentioned above need to be addressed. The fiber-optic link should be replaced with a wireless link as the technology becomes available. A different regulator and flow meter to allow greater flow rate to the air bearing will allow a larger air gap, reducing the possibility of grounding. New batteries and power converters will allow for a second CPU board and more digital electronics.

8.2.2 Adaptive Control and System Identification

All computed torque schemes are model referenced, requiring knowledge of system inertial parameters. While this is a reasonable assumption for a baseline design, large changes in inertial parameters could affect performance. For example, if the robot were to fetch a large tool, the changes in center of mass location could cause sufficient errors in the control law for it to miss the target. The robustness of the system would be improved if an adaptive identification of inertial parameters were incorporated into the control law.

8.2.3 Locomotion in Three Dimensions

The methods presented in this thesis readily extend to three dimensions. However, without experimentation, it is unclear how well these ideas can be implemented on a free-flying robot. Some technical issues that need to be addressed are:

1. Three dimensional sensing systems that will locate the robot, endpoint, and target position and orientation in real-time. Although three dimensional vision systems do exist, they are significantly more difficult to implement than planar systems. Also, vision systems tend to give very poor velocity information, because pixel information is quantized. Combining low cost INS sensors such as laser gyros with differential GPS could provide centimeter position information with smooth velocity estimates.

2. The exponential rise in performance for microprocessors has not been paralleled by a similar rise in performance for space-qualified processors. For a variety of reasons, including limited market and bureaucratic inertia, industry has been slow to implement the next-generation 32 bit space-qualified computer. This could be one of the major impedements to testing many new ideas and algorithms in space. Current space-qualified microprocessors are one to two orders of magnitude too slow, with onboard memory capabilities four to ten times too small.

8.2.4 Optimal Trajectory Generation

The fifth-order momentum trajectories were generated based upon the heuristic argument of modelling the robot as a point mass and accelerating that point in a desired direction. It is possible that another trajectory, having the same final momentum state, would be better. This would involve solving the Hamilton-Jacobi equations for a nonlinear system with limited actuators and geometric constraints. Because the problem is not convex, there is no guarantee of achieving global minimization. One of the features of computed torque is that it is trajectory-error based, so that any twice-differentiable function can be incorporated into the control law.

Another issue that needs further study is finding trajectories to take advantage of the ability of the robot to reorient itself without changing its angular momentum. As was shown in Chapter 7, the change in attitude is related *geometrically* to the path of shape changes. Finding optimal paths that cause large attitude changes for "small" shape change would be useful. It may turn out that reaction wheels or CMG's would be more practical for orientation control than relying on shape changes. On the other hand, some combination of shape changes and reaction torques may be better for some problems. This is a very fertile area with interesting theoretical as well as practical implications.

8.2.5 Multiple Vehicle Cooperation

Many tasks that are difficult to perform with one robot may be easier to accomplish with two or more robots. For example, assembling a large space structure composed of long slender beams is not amenable to being accomplished by only a single robot. Coordinating and managing multiple robots in an efficient manner is an interesting problem. Issues that need to be addressed include collision avoidance, distributive control strategies, parallel processing, and momentum management of multiple vehicles. For example, if two or more vehicles were to attach themselves to a free beam, it might be desirable to achieve zero net change in the beam's momentum. Using *system* momentum control, the momentum of *each* vehicle could be controlled at "pushoff" time so that when both robots attach themselves to the beam, the net momentum of the system is zero.

Appendix A

Calibration

Automated and semi-automated calibration procedures were developed for the joint angle sensors, pseudo-rate joint signals, angular rate sensor, and motor torque outputs. These routines collect data to be processed by Pro-Matlab, and provide a nice environment for data collection and analysis. This appendix describes how the various components of a free-flying robot are measured and calibrated.

A.1 Joint Angle Sensor Calibration

Calibrating joint angles on a free-flying robot is considerably harder than on a fixed-base robot. Most robotic applications desire knowledge of the manipulator endpoint, and so by fixing the endpoint in inertial space, one can back out the joint angles (assuming no redundancy) from inverse kinematics. For a free-flying robot, fixing the endpoint of the arms does not uniquely determine the arm's joint angles, since there are extra degrees of freedom in the base.

To circumvent this problem, a calibration fixture was constructed that could be attached to the robot in a repeatable manner. The fixture consisted of 32 holes drilled in a 1/8" aluminum sheet covering most of the workspace. A steel peg is placed through the tip of the arm to center it over a desired hole. Originally, the plate was designed such that only one joint at time changed angle as the arm moved in an arc over a subset of calibration holes. Only when the plate is attached perpendicular to the robot's axis of symmetry would this happen. Therefore, before any data was taken, one could be sure that the calibration plate was attached in a known and repeatable manner.

The joint angle calibration program prompts the user to position the tip in a numbered hole. It then uses inverse kinematics to calculate the actual joint angles (shoulder and elbow) for each location. A user specified number (usually 250) of measured points are

averaged at each location, and stored in a file with the actual measurement. A linear regression least-squares fit is performed to yield scale factor and offset. To further reduce the nonlinear effects of the sensor, a forth order polynomial fit is performed on the output of the least-squares fit.

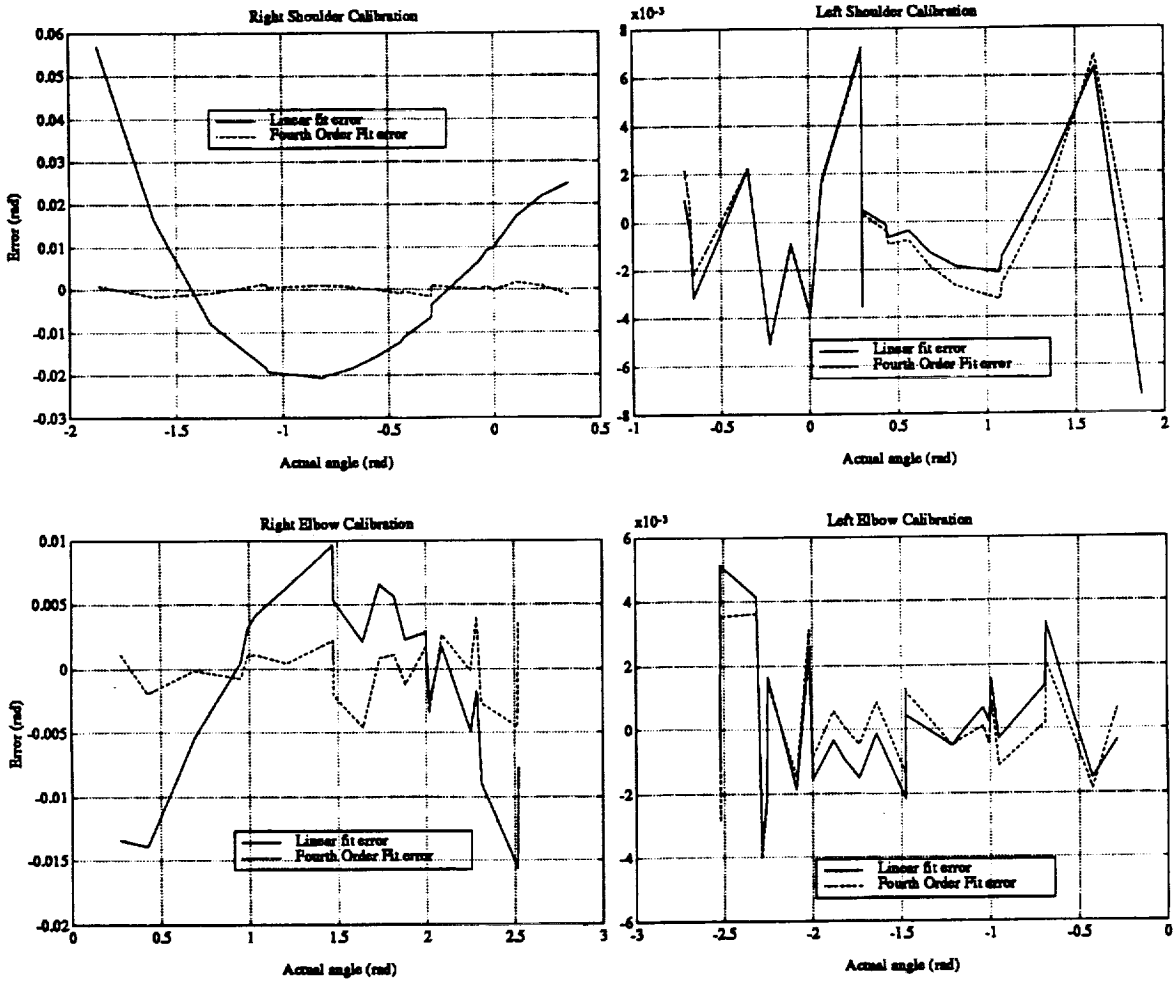

Figure A.1: **Joint Calibration**

The least-squares data was filtered through a fourth order polynomial to further reduce errors due to nonlinearities in the sensor.

A.2 Joint Velocity Sensor Calibration

After all the joint angles are calibrated, joint velocity calibration is performed. This is a completely automated procedure, whereby the arms are slewed through the workspace

under PD control while the velocity signals are integrated. By comparing the integrated velocity signal with the measured slew angle, the scale factor and offset for the pseudo-differentiated rate signal can be computed.

A.3 Model Parameter Measurements

Control methods such as computed torque and momentum control are model based and depend upon an accurate knowledge of the dimensional parameters of the plant. These include mass, center of mass location, joint location, and moment of inertia for each body. The masses of each body were weighted on an electronic scale to approximately 0.1 gram, while all lengths were machined to a tolerance of ± 0.005 inches. The location of the mass center was determined by balancing the object on two knife edges, and measuring the resulting force under each end. A simple static force/torque equilibrium calculation yields the position of the mass center.

Inertias for all the bodies except the base were measured with a trifilar pendulum, described in Appendix A of [13]. Briefly, the apparatus consists of an aluminum triangular plate, suspended from the ceiling by three wires. After the instrument is calibrated, a test object of mass m is placed on the plate such that the object's mass center and the plate's center are coincident. For small displacements, the period of oscillation, ω, is related to the moment of inertia, I, by the equation:

$$\omega = \sqrt{\frac{mgl^2}{IL}}$$

where l is the distance from the center of the plate to the suspension wires, L is the length of the support wires, and g is the acceleration of gravity. A good estimate of the inertia is obtained by averaging over then periods of oscillation. This procedure was automated by Stan Schneider and Larry Pfeffer, and is described in [29]. Comparison between measured inertias and those predicted by the CAD software package IDEAS [33] agree to 1%.

The base inertia was measured by a different means because its size and weight were too large for the trifilar pendulum. The base was floated with the arms set in a nominal tuck position. The moment of inertia of the system about the mass center can be derived from:

$$\dot{\theta}^2 = \frac{2T}{I}\theta$$

where θ and $\dot{\theta}$ are the angular position and rate of the base, T is the torque applied by the momentum wheel motor, and I is the inertia of the system. Knowing the inertia of the arms, one can derive the moment of inertia of the base. Using this method, it is estimated that the base inertia is known to about 7%.

Appendix B

Air Bearing

The following is a simplified derivation of an air bearing. Readers interested in a more detailed treatment on the subject should see Rehsteiner [25].

When solving most fluid mechanics problems, certain assumptions are made in order to solve the nonlinear Navier-Stokes equations. In this problem, we wish to derive the equations for an air bearing. The geometry of the problem consists of a source between two flat plates of radius r_o, with the top plate having a plenum radius r_i. It is assumed that the

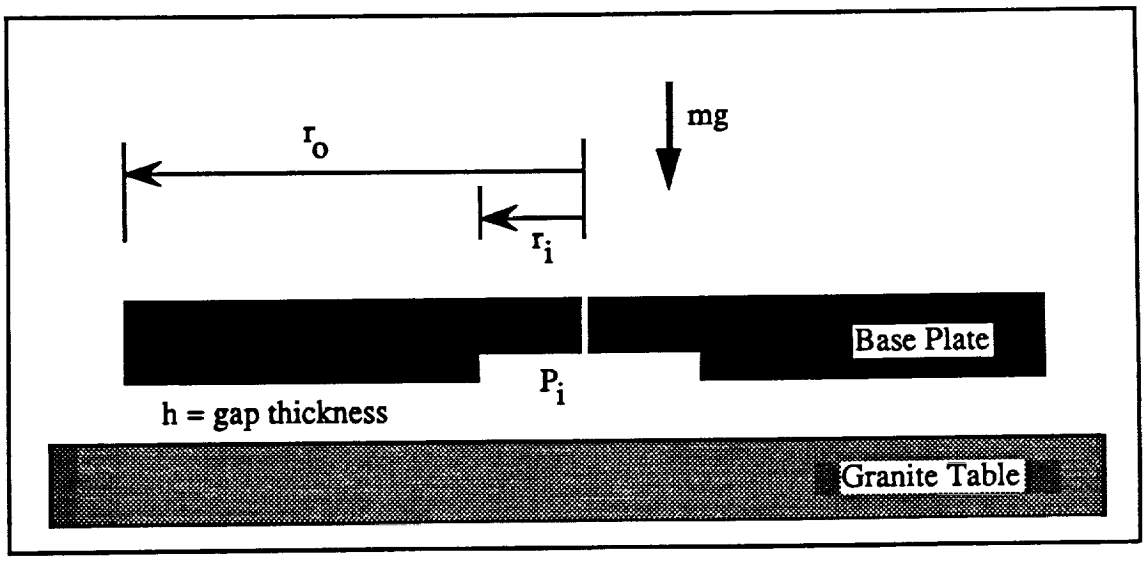

Figure B.1: **Air Bearing**

fluid is Newtonian (μ_0 = constant) and incompressible (ρ_0 = constant). Furthermore, based

on the geometry, we assume axisymmetric steady flow. These assumptions correspond to

$$\begin{aligned} \frac{\partial}{\partial t} &= 0 \qquad \text{steady flow} \\ u_z = u_\theta &= 0 \\ \frac{\partial}{\partial \theta} &= 0 \quad \text{axisymmetric flow} \\ \frac{\partial P}{\partial z} &= 0 \quad \text{neglect gravity} \end{aligned}$$

With these assumptions, the incompressible Navier-Stokes equations [35] can be formulated in cylindrical coordinates in the radial direction as:

$$\frac{1}{r}\frac{\partial}{\partial r}(ru_r) + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z} = 0 \tag{B.1}$$

$$u_r\frac{\partial u_r}{\partial r} = -\frac{1}{\rho_0}\frac{\partial P}{\partial r} + 2\nu_0\left[\frac{\partial^2 u_r}{\partial r^2} + \frac{1}{r}\frac{\partial u_r}{\partial r} - \frac{u_r}{r^2} + \frac{1}{2}\frac{\partial^2 u_r}{\partial z^2}\right] \tag{B.2}$$

Equation B.1, the conservation of mass equation, simplifies to:

$$\frac{1}{r}\frac{\partial}{\partial r}(ru_r) = 0 \tag{B.3}$$

$$u_r = \frac{f(z)}{r} \tag{B.4}$$

where $f(z)$ is an unknown function of z yet to be determined. Substituting this result into Equation B.2, the momentum equation, and simplifying yields:

$$\frac{1}{\rho_0}\frac{\partial P}{\partial r} = \frac{f(z)^2}{r^3} + \frac{\nu_0}{r}\frac{\partial^2 f(z)}{\partial z^2}$$

The first term on the right corresponds to momentum transfer due to inertial forces. Because the flow is laminar and slow, this term can be neglected, leaving just the second and dominant term. Physically, the equation relates the radial pressure drop due to the viscous force on the fluid. Integrating the above equation with respect to z yields:

$$\frac{1}{\rho_0}\frac{\partial P}{\partial r}\frac{z^2}{2} + C_1(r)z + C_2(r) = \frac{\nu_0}{r}f(z) \tag{B.5}$$

Where $C_1(r)$ and $C_2(r)$ are two unknow functions of r. If one now considers the case of a stationary base, the boundary conditions become $f(h/2) = f(-h/2) = 0$ which is the "no slip" condition at the two surfaces. Recall that h is the gap or thickness of the air-bearing. Substituting the boundary conditions into Equation B.5 gives:

$$\begin{aligned} C_1(r) &= 0 \\ C_2(r) &= \frac{-1}{2\rho_0}\frac{\partial P}{\partial r}\frac{h^2}{4} \\ f(z) &= \frac{r}{2\mu_0}\frac{\partial P}{\partial r}\left[z^2 - \frac{h^2}{4}\right] \end{aligned}$$

The last equation is a function of z only, imposing the condition:

$$r\frac{\partial P}{\partial r} = C_3 \quad \text{a constant}$$
$$P(r) = C_3 \ln(r/r_o) \quad r_i \leq r \leq r_o$$

For the air bearing to support the weight of the robot [1]

$$\begin{aligned} mg &= \int P dA \\ &= \int_0^{2\pi} \int_{r_i}^{r_o} C_3 \ln(r/r_o) r dr d\theta + \pi C_3 r_i^2 \ln(r_i/r_o) \\ &= \pi C_3 \left[\frac{r_i^2 - r_o^2}{2}\right] \end{aligned}$$

from which we can solve for the constant C_3. The velocity profile can now shown to be:

$$u_r = \frac{mg}{\mu_0 \pi (r_i^2 - r_o^2) r}\left[z^2 - \frac{h^2}{4}\right]$$

One last quantity of interest is Q, the flow rate. This can be calculated as:

$$\begin{aligned} Q &= \int_{-h/2}^{h/2} \int_0^{2\pi} u_r r d\theta dz \\ &= -\frac{\pi C_3 h^3}{6\mu_0} \end{aligned} \tag{B.6}$$

Equation B.6 shows why it is so difficult to increase the gap height of the air bearing: the flow rate is proportional to the cube of the gap thickness.

[1] It is also assumed that the pressure under the plenum is constant with a value of $P(r_i)$

Bibliography

[1] Harold L. Alexander. *Experiments in Control of Satellite Manipulators.* PhD thesis, Stanford University, Department of Electrical Engineering, Stanford, CA 94305, December 1987.

[2] Richard Bogen and et. al. *MACSYMA Reference Manual.* Computer Aided Mathematics Group of Symbolics, 8 New England Executive Park, East, Burlington, MA 01803, 13 edition, November 1988.

[3] Arthur E. Bryson, Jr. and Yu-Chi Ho. *Applied Optimal Control: Optimization, Estimation, and Control.* Hemisphere Publishing Corporation, 1025 Vermont Ave., N.W., Washington, D.C., 1975. Revised Printing.

[4] Robert H. Cannon, Jr., Marc Ullman, Ross Koningstein, Stan Schneider, Warren Jasper, Roberto Zanutta, and William Dickson. NASA Semi-Annual Report on Control of Free-Flying Space Robot Manipulator Systems. Semi-Annual Report 7, Stanford University Aerospace Robotics Laboratory, Stanford, CA 94305, August 1988.

[5] Craig R. Carignan. *Control Strategies For Manipulating Payloads In Weightlessness with a Free-Flying Robot.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, September 1987.

[6] John J. Craig. *Introduction to Robotics Mechanics and Control.* Addison-Wesley, Reading, MA, 1986.

[7] T. Dwyer. Exact nonlinear control of large angle rotational maneuvers. *IEEE Transactions on Automatic Control*, 29(9):769–774, September 1984.

[8] Tamar Flash and Neville Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. A. I. Memo 786, MIT Artificial Intelligence Laboratory, November 1984.

[9] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems.* Addison-Wesley, Reading, MA, June 1986.

[10] William J. Gilbert. *Modern Algebra with Applications.* John Wiley & Sons, New York, NY, 1976.

[11] Francis B. Hildebrand. *Methods of Applied Mathematics*, chapter 2, pages 119–138. Prentice-Hall, Inc., Englewood Cliffs, NJ, second edition, 1965.

[12] Walter Hohmann. *Die Erreichbarkeit Der Himmelskörper*, chapter 3, page 55. R. Oldenbourg, München und Berlin, 1925.

[13] Michael G. Hollars. *Experiments in End-Point Control of Manipulators with Elastic Drives.* PhD thesis, Stanford University, Department of Aeronautics and Astronautics, Stanford, CA 94305, May 1988. Also published as SUDAAR 568.

[14] Thomas R. Kane, M.R. Headrick, and J.D. Yatteau. Experimental investigation of an astronaut maneuvering scheme. *Journal of Biomechanics*, 5:313–320, 1972.

[15] Thomas R. Kane and David A. Levinson. *Dynamics: Theory and Application.* McGraw-Hill Series in Mechanical Engineering. McGraw-Hill, New York, NY, 1985.

[16] Thomas R. Kane, Peter W. Likins, and David A. Levinson. *Spacecraft Dynamics.* McGraw-Hill, New York, NY, 1983.

[17] Thomas R. Kane and M.P. Scher. Human self-rotation by means of limb movements. *Journal of Biomechanics*, 3:39–49, 1970.

[18] Oussama Khatib. A unified approach to motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, RA-3(1), February 1987.

[19] Raymond H. Kraft. *Experiments in End-Point Control of a Flexible Robot with a Mini-Manipulator.* PhD thesis, Stanford University, Department of Aeronautics and Astronautics, Stanford, CA 94305, May 1989. Also published as SUDAAR 581.

[20] Richard W. Longman, R. E. Lindberg, and M. F. Zedd. Satellite mounted robot manipulators - new kinematics and reaction moment compensation. In *AIAA Guidance, Navigation and Control Conference*, pages 278–290, Snowmass, CO, August 19-21 1985. Technical Papers.

[21] H. Makino and N. Furuya. SCARA robot and its family. In *Proceedings of the 3rd International Conference on Assembly Automation and 14th IPA Conference*, pages 433–444, Stuttgart, Germany, May 1982.

[22] M. S. Mujtaba. Discussion of trajectory calculation methods. In Thomas O. Binford, editor, *Exploratory Study of Computer Integrated Assembly Systems.* Stanford University, June 1977. Computer Science Department Report STAN-CS-76-568.

[36] Christopher R. Uhlik. *Experiments in High-Performance Nonlinear and Adaptive Control of a Two-Link, Flexible-Drive-Train Manipulator.* PhD thesis, Stanford University, Department of Electrical Engineering, Stanford, CA 94305, May 1990.

[37] Charles W. Wampler. *Computer Methods in Manipulator Kinematics, Dynamics, and Control: A Comparative Study.* PhD thesis, Stanford University, Department of Mechanical Engineering, Stanford, CA 94305, December 1984.

[38] James R. Wertz, editor. *Spacecraft Attitude Determination and Control*, volume 73 of *Astrophysics and Space Science Library.* D. Reidel Publishing Company, 190 Old Derby Street, Hingham, MA 02043, 1978.